



Titre: Développement d'un module de réglage automatique d'un
Title: compensateur Pid Dual Loop

Auteur: Ernesto Cornieles
Author:

Date: 1998

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Cornieles, E. (1998). Développement d'un module de réglage automatique d'un
Citation: compensateur Pid Dual Loop [Thèse de doctorat, École Polytechnique de
Montréal]. PolyPublie. <https://publications.polymtl.ca/8654/>

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/8654/>
PolyPublie URL:

**Directeurs de
recherche:**
Advisors:

Programme: Non spécifié
Program:

UNIVERSITÉ DE MONTRÉAL

DÉVELOPPEMENT D'UN MODULE
DE RÉGLAGE AUTOMATIQUE
D'UN COMPENSATEUR PID DUAL LOOP

ERNESTO CORNIELES
DÉPARTEMENT DE GÉNIE ÉLECTRIQUE
ET DE GÉNIE INFORMATIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

THÈSE PRÉSENTÉE EN VUE DE L'OBTENTION
DU DIPLÔME DE PHILOSOPHIAE DOCTOR (Ph.D.)
(GÉNIE ÉLECTRIQUE)
JUILLET 1998

© Ernesto Cornieles, 1998.



National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-37375-4

Canada

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Cette thèse intitulée:

DÉVELOPPEMENT D'UN MODULE
DE RÉGLAGE AUTOMATIQUE
D'UN COMPENSATEUR PID DUAL LOOP

Présentée par: CORNIELES Ernesto

en vue de l'obtention du diplôme de: Philosophiae Doctor

a été dûment acceptée par le jury d'examen constitué de:

M. O'SHEA Jules, D.Ing., président

M. De SANTIS Romano M., Ph.D., membre et directeur de recherche

M. HURTEAU Richard, D.Ing., membre

M. SAAD, Maarouf, Ph.D., membre externe

À:

Dalia, Daniel et

Barbara

REMERCIEMENTS

J'aimerais exprimer ma reconnaissance à mon directeur de recherche, le professeur Romano M. De Santis, pour m'avoir judicieusement guidé tout au long du projet de même que pour son appui financier; au professeur Jules O'Shea, pour m'avoir aidé à plusieurs occasions entre autres dans l'implantation du banc d'essai; au professeur Richard Hurteau, pour m'avoir fourni les sous-routines d'acquisition de données.

Je tiens aussi à remercier Éric Coneim et Geoffrey Mavel pour avoir collaboré avec moi à la mise au point du banc d'essai et à l'identification des composantes. Également, mes sincères remerciements à Vincent Dimitriou, Christophe Bougeret et Angel Ruiz pour avoir collaboré avec moi à l'étude théorique du système et à la conception d'un compensateur neuronal.

Je tiens également à exprimer ma reconnaissance à M. Yves Léonard, technicien de la section d'automation et systèmes pour m'avoir aidé à surmonter les nombreuses difficultés pratiques rencontrées dans la réalisation de ce projet.

Je remercie également la Universidad del Zulia et Conicit qui m'ont attribué une bourse d'étude.

Finalement, j'adresse mes remerciements les plus chaleureux à mon épouse Dalia et mes enfants Daniel et Barbara pour leur compréhension pendant mes études à l'École Polytechnique.

RÉSUMÉ

Cette thèse concerne l'étude des compensateurs de procédés industriels, de type Proportionnel, Intégral, Dérivé (PID). Elle poursuit un triple objectif. D'abord, le développement d'une étude comparative des procédures de réglage des gains d'une nouvelle classe de compensateurs PID (PID Dual Loop) avec les procédures de réglage des gains de compensateurs PID actuellement utilisées dans l'industrie. Ensuite, une amélioration des procédures de réglage de la nouvelle classe de compensateurs PID Dual Loop qui rendra ces mêmes procédures automatiques. Finalement, une validation expérimentale de l'atteinte des deux premiers objectifs moyennant l'utilisation d'un banc d'essai de type industriel que l'on a mis au point à ces fins. Le banc d'essai est constitué d'un réservoir d'alimentation d'eau dont il faut contrôler la pression (niveau) et la température. La mise au point a impliqué l'exécution de plusieurs phases: le choix, l'implantation et la mise en opération des composantes hydrauliques; le choix et l'implantation de l'instrumentation des différentes interfaces; le développement des sous-routines pour l'implantation des compensateurs et le traitement des données; l'identification des différentes composantes et du modèle du procédé. L'étude comparative des procédures de réglage nous a amené à comparer les performances du nouveau compensateur PID avec les compensateurs classiques et avec celles des gains réglés en utilisant les techniques de Ziegler-Nichols, Cohen-Coon, Morari et la technique d'optimisation ITAE (Integral Time Absolute Error). Ces performances ont aussi été comparées avec celles de la nouvelle classe de régulateurs modernes basés sur l'emploi

de la théorie des réseaux neuronaux. L'implantation des procédures de réglage des gains d'un compensateur PID Dual Loop a été caractérisée par le développement d'un algorithme d'autoréglage et d'un logiciel de commande adaptative en temps réel.

ABSTRACT

This thesis deals with the study of industrial processes of controllers, Proportional, Integral, Derived types (PID), pursuing a triple objective. First, the development of a comparative study of procedures in tuning of gains of a new class of PID controllers (PID Dual Loop) with tuning of gains of a PID controller currently used by the industry. Second, an improvement of the tuning procedures of the new class of PID Dual Loop controller that will achieve some automatic procedures. Finally, an experimental validation of the achievements of the two first objectives using an industrial type test bench. The test bench is formed by a water-feed-tank that must control the pressure (level) and temperature. The tuning includes the execution of several phases: the choice, the implementation and operation of hydraulic components; the choice and instruments implementation of different interphases; the development of sub-routines by controllers implementations and data processing; the identification of different components and model process. The comparative study of tuning procedures takes us to compare the new PID controller performance with classic controllers and with those of tuned gains using the Ziegler-Nichols, Cohen-Coon techniques and the ITAE (Integral Time Absolute Error) optimisation technique. These performances have as well been compared with those of the new class of modern regulators based on the neural control theory. The implementation of gain tuning procedures of a PID Dual Loop controller has been characterized by the development of an auto-tuning algorithm and a software of adaptive command in real time.

TABLE DES MATIÈRES

DÉDICACE	iv
REMERCIEMENTS	v
RÉSUMÉ	vi
ABSTRACT	viii
TABLE DES MATIÈRES	ix
LISTE DES TABLEAUX	xv
LISTE DES FIGURES	xvi
LISTE DES SYMBOLES	xx
LISTE DES ANNEXES	xxiv
INTRODUCTION	1
0.1 Le compensateur PID	1
0.2 Objectifs de notre recherche.....	4
0.3 Originalité.....	5
0.4 Organisation de la thèse.....	6
 CHAPITRE I: STRUCTURE ET RÉGLAGE DES GAINS	
D'UN COMPENSATEUR PID DUAL LOOP	7
1.1 Introduction	7
1.2 Compensateur PID Dual Loop	7
1.3 Deux propriétés caractéristiques du PID Dual Loop.....	9

1.4	Procédure de réglage automatique des gains.....	15
1.4.1	Cas monovariable.....	16
1.4.2	Première étape du réglage	17
1.4.3	Seconde étape du réglage	23
1.4.4	Cas multivariable.....	25
1.5	Organigramme de l'algorithme de réglage automatique	26
 CHAPITRE II: IMPLANTATION D'UN BANC D'ESSAI		27
2.1	Introduction	27
2.2	Description générale du banc d'essai	27
2.2.1	Réservoir.....	27
2.2.2	Interaction entre le système et le compensateur	30
2.2.3	Commande des vannes d'eau chaude et d'eau froide.....	31
2.2.4	Conditions initiales du système	32
2.2.5	Modèle du système	34
2.2.6	Logiciel d'asservissement	35
 CHAPITRE III: ÉTUDE COMPARATIVE EXPÉRIMENTALE DES		
COMPENSATEURS PID ET PID DUAL LOOP		38
3.1	Introduction	38
3.2	Différentes techniques de réglage des compensateurs PI classique et PID Dual Loop.....	38

3.2.1	Réglage du compensateur PI classique.....	38
3.2.2	Réglage du compensateur PID Dual Loop	39
3.3	Résultats expérimentaux.....	40
3.3.1	Logiciel d'asservissement du niveau et de la température	41
3.4	Test sur le contrôle du niveau seul	44
3.4.1	Asservissement du niveau seul.....	44
3.4.2	Test de la robustesse.....	47
3.5	Test sur le contrôle de la température seule	48
3.5.1	Asservissement de la température seule	48
3.5.2	Test de la robustesse.....	52
3.6	Test des deux boucles scalaires	54
3.6.1	Asservissement simultané du niveau et de la température	54
3.6.2	Test de la robustesse.....	56
3.7	Difficultés techniques.....	58
3.8	Conclusion sur les expériences.....	58

CHAPITRE IV: ÉTUDE EXPÉRIMENTALE DU COMPENSATEUR

PID DUAL LOOP AVEC UN MODULE D'ADAPTATION

AUTOMATIQUE..... 61

4.1	Introduction	61
4.2	Rappel des structures du banc d'essai et du compensateur PID Dual Loop	62
4.2.1	Structure générale de l'asservissement.....	62

4.3	Résultats expérimentaux.....	63
4.3.1	Niveau.....	63
4.3.2	Température.....	72
4.3.3	Niveau et température.....	77
4.4	Difficultés techniques.....	84
4.5	Conclusions générales sur les expériences.....	85

CHAPITRE V: ASSERVISSEMENT DU NIVEAU ET DE LA

TEMPÉRATURE AU MOYEN D'UN

COMPENSATEUR NEURONAL 87

5.1	Introduction	87
5.2	Le compensateur neuronal.....	87
5.3	Structure du compensateur neuronal	89
5.3.1	Le neuro-compensateur	89
5.3.2	Apprentissage "hors ligne"	90
5.3.3	Apprentissage en ligne	92
5.3.4	Choix des entrées.....	94
5.3.5	Algorithme d'apprentissage en ligne	95
5.4	Algorithme de "Back-propagation"	95
5.5	Résultats expérimentaux.....	98
5.5.1	Logiciel du compensateur neuronal.....	101
5.5.2	Contrôle du niveau	104

5.5.2.1	Contrôle du niveau seul.....	105
5.5.2.2	Contrôle du niveau en présence d'une perturbation.....	106
5.5.2.3	Robustesse du compensateur en présence d'une perturbation et d'un retard.....	108
5.5.3	Contrôle de la température	109
5.5.3.1	Contrôle de la température seule.....	110
5.5.3.2	Contrôle de la température en présence d'une perturbation	112
5.5.3.3	Robustesse du compensateur en présence d'une perturbation et d'un retard.....	113
5.5.4	Test des deux boucles scalaires	115
5.5.4.1	Contrôle des deux boucles en présence d'une perturbation	117
5.5.4.2	Robustesse du compensateur en présence d'une perturbation et d'un retard.....	119
5.6	Conclusion sur les expériences.....	121

CHAPITRE VI: COMPARAISON EXPÉRIMENTALE D'UN

COMPENSATEUR NEURONAL ET D'UN

	COMPENSATEUR PID DUAL LOOP.....	122
6.1	Introduction	122
6.2	Résultats expérimentaux.....	122

6.3	Logiciel d'asservissement du niveau et de la température	123
6.4	Test sur le contrôle du niveau seul.....	124
6.4.1	Asservissement du niveau seul.....	124
6.5	Test sur le contrôle de la température seule.....	126
6.5.1	Asservissement de la température seule	126
6.6	Test des deux boucles scalaires	127
6.6.1	Asservissement simultané du niveau et de la température	127
6.7	Conclusion générale sur les expériences	129
 CONCLUSION GÉNÉRALE.....		130
RÉFÉRENCES BIBLIOGRAPHIQUES		133
ANNEXES.....		145

LISTE DES TABLEAUX

Tableau 3.1: Paramètres des compensateurs PI et PID Dual Loop pour la régulation de niveau	45
Tableau 3.2: Paramètres des compensateurs PI et PID Dual Loop pour la régulation de température.....	49
Tableau 4.1: Résultats de la première expérience concernant le niveau	64
Tableau 4.2: Résultats de la deuxième expérience concernant le niveau.....	69
Tableau 4.3: Résultats de l'expérience concernant la température	73
Tableau 4.4: Résultats de l'expérience concernant le niveau et la température	79

LISTE DES FIGURES

Figure 1.1:	Structure du compensateur PID Dual Loop.....	8
Figure 1.2:	Schéma-bloc d'un asservissement.....	9
Figure 1.3:	Schéma-bloc de l'autoréglage (monovariable).....	16
Figure 1.4:	Schéma-bloc de l'asservissement pendant la première étape de réglage.....	18
Figure 1.5:	Schéma-bloc de l'autoréglage (multivariable)	25
Figure 2.1:	Représentation du système complet	29
Figure 2.2:	Interactions entre système et compensateur	31
Figure 2.3:	Schématisation d'une vanne.....	32
Figure 2.4:	Organigramme du logiciel d'asservissement.....	37
Figure 3.1:	Structure du PID Dual Loop.....	41
Figure 3.2:	Structure générale de l'asservissement.....	42
Figure 3.3:	Régulation de niveau. Comparaison de différentes techniques de réglage	46
Figure 3.4:	Tests de la robustesse avec différents réglages	48
Figure 3.5a:	Régulation de température. Comparaison de différentes techniques de réglage	50
Figure 3.5b:	Régulation de température. Comparaison de différentes techniques de réglage	50

Figure 3.6:	Commandes des vannes avec les réglages de Rivera-Morari et de Cohen-Coon	52
Figure 3.7:	Comparaison de la robustesse avec deux compensateurs différents	53
Figure 3.8:	Fonctionnement simultané des deux asservissements.....	55
Figure 3.9:	Niveau et température face à une perturbation.....	57
Figure 4.1:	Structure générale de l'asservissement.....	62
Figure 4.2:	Résultat global du réglage automatique (niveau et commandes) des 7 tests.....	65
Figure 4.3:	Évolution de la réponse lors de la première étape du réglage (niveau).....	66
Figure 4.4:	Évolution de la robustesse avec l'augmentation de K13 (niveau)	67
Figure 4.5:	Résultat global du réglage automatique (niveau) des 15 tests.....	70
Figure 4.6:	Évolution de la réponse lors de la première étape du réglage (niveau).....	70
Figure 4.7:	Évolution de la robustesse avec l'augmentation de K13 (niveau)	71
Figure 4.8:	Résultat global du réglage automatique (température et commandes).....	74
Figure 4.9:	Évolution de la réponse lors de la première étape du réglage (température)	75
Figure 4.10:	Évolution de la robustesse avec l'augmentation de K23 (température)	76

Figure 4.11: Résultat global du réglage automatique (commandes, niveau et température).....	80
Figure 4.12: Évolution de la réponse en niveau lors de la première étape du réglage	81
Figure 4.13: Évolution de la réponse en température lors de la première étape du réglage	81
Figure 4.14: Évolution de la robustesse avec l'augmentation de K13 et K23 (niveau et température).....	82
Figure 4.15: Comportement du système lorsque $K11=0.85$, $K12=12.2$, $K13=0.006$, $A11=37.3$, $A12=347$, $K21=2.42$, $K22=35$, $K23=0.05$, $A21=75$, $A22=1416$ (avec un retard de 30s et une perturbation d'un volt sur la commande d'eau froide).....	86
Figure 5.1: Structure "Indirect Learning Architecture"	88
Figure 5.2: Neuro-compensateur. Structure d'apprentissage.....	90
Figure 5.3: Génération du fichier entrée/sortie	91
Figure 5.4: Neuro-émulateur. Structure d'apprentissage	93
Figure 5.5: Structures du neuro-émulateur et du neuro-compensateur.....	96
Figure 5.6: Algorithme du logiciel de contrôle.....	103
Figure 5.7: Test de régulation du niveau.....	105
Figure 5.8: Contrôle du niveau en présence d'une perturbation	107
Figure 5.9: Robustesse du compensateur en présence d'une perturbation et d'un retard.....	108

Figure 5.10:	Contrôle de la température	111
Figure 5.11:	Contrôle de la température en présence d'une perturbation.....	112
Figure 5.12:	Robustesse du compensateur en présence d'une perturbation et d'un retard.....	114
Figure 5.13:	Contrôle des deux boucles scalaires	116
Figure 5.14:	Contrôle des deux boucles en présence d'une perturbation	118
Figure 5.15:	Robustesse du compensateur en présence d'une perturbation et d'un retard.....	120
Figure 6.1:	Régulation de niveau. Comparaison de deux compensateurs	125
Figure 6.2:	Comparaison de la réponse de la température avec deux compensateurs	126
Figure 6.3:	Comparaison de la réponse du niveau et de la température avec les deux compensateurs.....	128

LISTE DES SYMBOLES

α	retard (en s)
α_r	rapport des commandes en tension pour le réglage des débits d'eau froide et d'eau chaude (sans unité)
α_1, α_2	coefficients du dénominateur de la fonction de transfert après une itération
α_{10}, α_{20}	coefficients du dénominateur de la fonction de transfert avant une itération
β_1, β_2	coefficients du découpleur pour la boucle de température
δ^x	erreur propagée à la couche X
Δw^x	variation des valeurs des poids de la couche X
ε_t	erreur d'apprentissage = référence - $\hat{y}(t)$
ε_u	erreur d'estimation du NC = $u(t) - \hat{u}(t)$
ε_y	erreur d'estimation du NE = $y(t) - \hat{y}(t)$
η	coefficient d'apprentissage
$\Gamma(s)$	fonction de transformation de la robustesse
τ	constante du temps (en s)
τ_{BF}	constante du temps en boucle fermée (en s)
τ_p	constante de temps du procédé (en s)
μ	coefficient de multiplication de la pulsation naturelle

μ_k	coefficient de multiplication de l'amortissement
ω_n	pulsation naturelle après une itération
ω_{n0}	pulsation naturelle avant une itération
ξ	amortissement après une itération
ξ_0	amortissement avant une itération
a_1, a_2	coefficients du dénominateur de l'approximation du second ordre
$A1, A2$	noms génériques des gains de la boucle externe du compensateur PID Dual Loop
$A11, A12$	gains de la boucle externe du compensateur de niveau
$A21, A22$	gains de la boucle externe du compensateur de température
$ampl$	amplitude de premier dépassement désirée à l'issue de la première étape de réglage
a_x	vecteur de sortie de la couche "X" d'un réseau de neurones
$depn$	dépassement du niveau (en V)
$dept$	dépassement de la température (en V)
f_x	fonction d'activation de la couche X
F_1	fonction de transfert consigne/mesure après une itération
F_2	fonction de transfert perturbation/mesure après une itération
F_{10}	fonction de transfert consigne/mesure avant une itération
F_{20}	fonction de transfert perturbation/mesure avant une itération
G_1, G_2	fonctions de transfert particulières
k_1, k_2	gains proportionnel et dérivatif après une itération

$K1, K2, K3, K4$	noms génériques des gains de la boucle interne du compensateur PID Dual Loop
k_{10}, k_{20}	gains proportionnel et dérivatif avant une itération
$K11, K12, K13, K14$	gains de la boucle interne du compensateur de niveau
$K21, K22, K23, K24$	gains de la boucle interne du compensateur de température
NC	réseau de neurones qui génère la commande pour le procédé (Neuro-compensateur)
NE	réseau de neurones qui copie la dynamique du procédé (Neuro- émulateur)
<i>oscil</i>	présence d'oscillations dans la réponse
<i>overshoot</i>	amplitude réelle de premier dépassement
P_s	consigne
P_t	perturbation ajoutée volontairement sur la commande pour tester la robustesse
P_v	mesure
$pv1, pv2$	signaux d'évaluation du niveau et de la température fournis par le système au compensateur (en V)
réf	consigne (en V)
réf1, réf2	consignes de niveau et de température (en V)
<i>t_{fpa}</i>	temps réel de premier dépassement de la réponse
<i>t_{pd}</i>	temps du premier dépassement désiré à l'issue de la première étape de réglage

$tpdn$	temps calculé du premier dépassement pour le niveau (en s)
$tpdt$	temps calculé du premier dépassement pour la température (en s)
$u(t)$	commande appliquée au procédé (en V)
$\hat{u}(t)$	commande estimée par le NC (en V)
$u1, u2$	signaux de commandes des vannes d'eau chaude et d'eau froide fournis par le compensateur du système (en V)
$U1, U2$	commandes en sortie de compensateur (en V)
$u11, u12$	commandes partielles des vannes d'eau chaude et d'eau froide issues du compensateur de niveau (en V)
$u21, u22$	commandes partielles des vannes d'eau chaude et d'eau froide issues du compensateur de température (en V)
w^x	poids (weights) entre les entrées de la couche X et les sorties de la couche précédente
$X1s$	consigne du niveau (en m)
Xc	vecteur d'entrée pour le neuro-compensateur
Xe	vecteur d'entrée pour le neuro-émulateur
$y(t)$	sortie du procédé (en V)
$\hat{y}(t)$	sortie du procédé estimée par le neuro-émulateur (en V)

LISTE DES ANNEXES

Annexe 1:	Photo du banc d'essai pour l'asservissement du niveau et de la température	145
Annexe 2:	Liste des composantes du banc d'essai	146
Annexe 3:	Schéma complet de l'interface (4).....	148
Annexe 4	Différentes interfaces du système complet.....	149
Annexe 5:	Organigramme du module de réglage automatique du compensateur PID Dual Loop	151
Annexe 6:	Organigramme du programme NEWGAIN.m	156
Annexe 7:	Techniques de réglage du PID classique	157
Annexe 8:	Organigramme de la solution du problème en utilisant le PID Dual Loop.....	161
Annexe 9:	Procédure de réglage des paramètres du compensateur PID Dual Loop	162
Annexe 10:	Programmes du module de réglage automatique du compensateur PID Dual Loop	164

INTRODUCTION

0.1 Le compensateur PID

Le compensateur proportionnel – intégral – dérivatif (PID) est le compensateur le plus utilisé dans le domaine des procédés industriels. L'utilisation du PID est due à sa robustesse et à sa simplicité (Hang, Aström et Ho, 1991; Aström et Hägglund, 1992; Voda et Landau, 1995). Mais le réglage des gains du compensateur PID représente un problème pour l'industrie: de longues durées de temps pour l'ajustement fin des gains et un coût économique et humain pour obtenir les meilleures performances (Vega, Prada et Aleixandre, 1991; Zhuang et Atherton, 1993). Une grande quantité de chercheurs ont traité ce problème. Plusieurs approches et techniques ont été proposées pour déterminer les paramètres d'un compensateur PID, telles que les méthodes de Ziegler-Nichols (1942) et Cohen-Coon (1953); ils ont utilisé la réponse en boucle ouverte et la réponse en fréquence en boucle fermée pour obtenir, d'une manière heuristique, le réglage d'un compensateur PID. Les méthodes de pôles dominants, le domaine de fréquence et la technique de placement de pôles ont été proposés par D'Azzo et Houpis (1981), Aström et Hägglund (1984, 1988), Hang, Aström et Ho (1991), Kuo (1995), Aström, Hang, Persson et Ho (1992). Beaucoup d'efforts ont été faits pour améliorer ces techniques et obtenir de meilleures performances (Aström et Hägglund, 1984; Hang, Aström et Ho, 1991; Vega Prada et Aleixandre, 1991; Voda et Landau, 1995; Zhuang et Atherton, 1993). Des algorithmes plus efficaces, par exemple l'algorithme pour la minimisation quadratique, ont été présentés (Gawthrop, 1986; Bortolotto, Desages et Romagnoli,

1989; Al-Saggaf, 1991; Ruano, Fleming et Jones, 1992; Zhuang et Atherton, 1993). L'algorithme d'optimisation basé sur des critères en fonction de la caractéristique de la réponse d'un système soumis à l'action d'un compensateur PID a été proposé pour obtenir, d'une manière optimale, les paramètres du compensateur (Al-Assadi et Al-Chalabi, 1987; Porter, Jones et McKeon, 1987; Aström et Hägglund, 1988; Cheung, 1988). La technique du relais pour obtenir les gains d'un compensateur PID de façon automatique a été proposée par Hägglund et Aström (1991).

Les algorithmes experts et l'autoréglage d'un compensateur PID sont présentés comme une nouvelle stratégie pour obtenir avec précision les gains et constantes de temps d'un compensateur PID (Hang et Sin, 1991; Aström, Hägglund et Wallenborg, 1993; Hang, Ho et Cao, 1994). La technique IMC (modèle de contrôle interne) a été présentée par Garcia et Morari (1982). L'approche permet d'obtenir, pour différents modèles, le correspondant IMC-PID (Rivera et Morari, 1986). Le prédicteur Smith a été utilisé pour obtenir une structure IMC-PID (Smith, 1957; Aström, Hang et Lim, 1994). Lorsque les paramètres d'un système à asservir sont soumis à des variations importantes, il est alors souvent nécessaire de modifier les valeurs des gains du compensateur PID. L'approche de conception d'un compensateur adaptatif avec modèle de référence se donne pour objectif d'améliorer le comportement du système ainsi conçu par la prise en charge des variations paramétriques du système (Slotine et Li, 1991). Dans beaucoup de systèmes de commande, il est important que l'opération du système soit insensible aux variations des paramètres avec les compensateurs

actuellement disponibles; cette exigence peut être difficile à satisfaire. En vue d'améliorer cette situation, de nombreux articles scientifiques ont attiré l'attention sur les avantages potentiels offerts par les compensateurs à structure variable (Slotine, 1984; Serfass, 1987; De Santis, 1989; Slotine et Li, 1991; Cho, Kato et Spilman, 1993). Une nouvelle structure du compensateur PID développée par De Santis en 1994 se base sur l'interprétation de la structure classique d'un contrôleur PID qui consiste à découpler parallèlement la satisfaction des contraintes de performances dynamiques et la robustesse. L'étude de sensibilité comparative est présentée dans les travaux de De Santis et Porter (1974), De Santis et Conan (1977), Thompson (1982), Garcia et Morari (1982), Gilliam et Vinante (1992) et Häggblom (1993). Le développement pour le découplage interne pour des systèmes multivariables a également été proposé (Bristol, 1966; Balchen, Lie et Solberg, 1988), de même que la linéarisation globale pour systèmes multivariables (Kravaris et Chung, 1987; Kravaris et Soroush, 1990; Seborg, Edgar et Mellichamp, 1989; Henson et Seborg, 1990; Nakamoto et Watanabe, 1991). Le compensateur industriel de la compagnie Foxboro® utilise un système expert, approche heuristique basée sur les travaux de Ziegler et Nichols, Bristol et Shinskey. Le compensateur utilise une carte EXACT (*Expert Adaptive Controller Tuning*) permettant à l'appareil de régler lui-même les paramètres (Aström et Hägglund, 1992). Le principe est basé sur l'identification du procédé et la reconnaissance de la forme de la réponse. Un autre compensateur industriel de la compagnie Satt Control Instruments® (Aström et Hägglund, 1992) permet le réglage automatique des gains du compensateur PID en utilisant l'estimation d'un point de la courbe de Nyquist et la

méthode du relais (Aström et Hägglund, 1984). L'autoréglage est introduit par la compagnie Texas Instrument® en 1984 en utilisant le système DDC (Système de contrôle numérique direct) pour la régulation d'un procédé industriel (Aström et Hägglund, 1992).

L'abondance des travaux techniques nous donne une bonne idée de la problématique pour obtenir le réglage des gains du compensateur PID (recherche des modalités plus rapides pour le réglage des paramètres, optimisation des performances concernant le comportement dynamique/statique, simplification des actions de mise en opération).

0.2 Objectifs de notre recherche

Cette recherche poursuit un triple objectif. D'abord, le développement d'une étude comparative des procédures de réglage des gains d'une nouvelle classe de compensateurs PID (PID Dual Loop) avec les procédures de réglage des gains de compensateurs PID actuellement utilisées dans l'industrie. Ensuite, une amélioration des procédures de réglage de la nouvelle classe de compensateurs PID Dual Loop qui rendra ces mêmes procédures automatiques. Finalement, une validation expérimentale de l'atteinte des deux premiers objectifs moyennant l'utilisation d'un banc d'essai de type industriel.

Le procédé en étude est caractérisé par un banc d'essai multivariable qui représente un prototype de procédé industriel. Le système est non linéaire (saturation, seuil, courbes

caractéristiques entrée/sortie non linéaires), présente un délai entre l'action de contrôle et son influence sur la variable contrôlée, il est soumis à des perturbations extérieures et à des variations de paramètres (Bristol, 1966; Balchen, Lie et Solberg, 1988; Barton, Chan et Perkins, 1991; Häggblom, 1993). Ces propriétés font en sorte que les techniques classiques de l'automatique ne sont pas nécessairement et facilement applicables au problème.

0.3 Originalité

Ce projet de recherche a contribué à développer des techniques et des algorithmes de réglage d'un compensateur PID Dual Loop. Un banc d'essai multivariable pour l'asservissement de niveau et de température d'un réservoir d'eau a été conçu et validé. Ce prototype industriel peut être utilisé à des fins pédagogiques et de recherche.

L'originalité de cette recherche a été la conception, l'implantation et la validation d'un module de réglage automatique en ligne du compensateur PID Dual Loop. Ce module a permis d'obtenir des résultats très satisfaisants pour l'asservissement de niveau et de température.

Les différents tests ou expériences qui ont servi à comparer les techniques ou procédures de réglage des compensateurs ont été réalisés avec les mêmes conditions d'opération que celles du banc d'essai. Ce banc d'essai, développé à la section d'automation et systèmes du département de génie électrique et de génie informatique

de l'École Polytechnique (Cornieles et Mavel, 1997), fut utilisé tout au long du projet; il s'agit d'un réservoir d'eau dont il faut réguler le niveau et la température.

0.4 Organisation de la thèse

Pour développer cette recherche et atteindre les objectifs, le sujet a été décomposé en plusieurs parties, soit: la conception et le développement d'un module de réglage automatique du compensateur PID Dual Loop en un ensemble de sous-problèmes à résoudre, une revue de la littérature concernant les compensateurs PID, un rappel des différentes théories des compensateurs, la mise au point du banc d'essai pour l'asservissement de niveau et de température d'un réservoir d'eau, l'étude théorique et expérimentale de différentes techniques de réglage des compensateurs PID et PID Dual Loop, l'étude comparative de différentes techniques de réglage et la mise au point du banc d'essai avec le module de réglage automatique des gains du compensateur PID Dual Loop. Le contenu de la thèse est réparti sur six chapitres: au chapitre I, on retrouve la structure du compensateur PID Dual Loop; au chapitre II, l'implantation d'un banc d'essai; au chapitre III, l'étude comparative expérimentale des compensateurs PID et PID Dual Loop; au chapitre IV, l'étude expérimentale du compensateur PID Dual Loop avec un module d'adaptation automatique; au chapitre V, la conception d'un compensateur neuronal sur le banc d'essai; et au chapitre VI, l'étude comparative du compensateur neuronal et du PID Dual Loop; finalement, une conclusion générale, les différentes annexes et les programmes utilisés pour les simulations et les différentes expériences.

CHAPITRE I

STRUCTURE ET RÉGLAGE DES GAINS

D'UN COMPENSATEUR PID DUAL LOOP

1.1 Introduction

L'objectif de ce chapitre est de présenter le compensateur PID Dual Loop (configuration particulière pour un système scalaire avec dynamique approximable pour un système du premier ou deuxième ordre). On présente également le développement d'un module de réglage automatique des gains du compensateur PID Dual Loop pour un système monovisible et pour un système multivariable.

1.2 Compensateur PID Dual Loop

La structure d'un compensateur de la famille PID Dual Loop (De Santis, 1994) est constituée de deux boucles comme on le présente à la figure 1.1.

Ce compensateur PID Dual Loop est basé sur l'interprétation de la structure classique d'un compensateur PID qui consiste, d'une part, à découpler parallèlement la satisfaction de performances dynamiques (boucle interne) constituée par les gains K_1 , K_2 et K_4 , et d'autre part, la robustesse (boucle externe) constituée par les gains K_3 , A_1 et A_2 . Cette structure permet d'obtenir le PID classique en prenant A_1 et A_2 nuls.

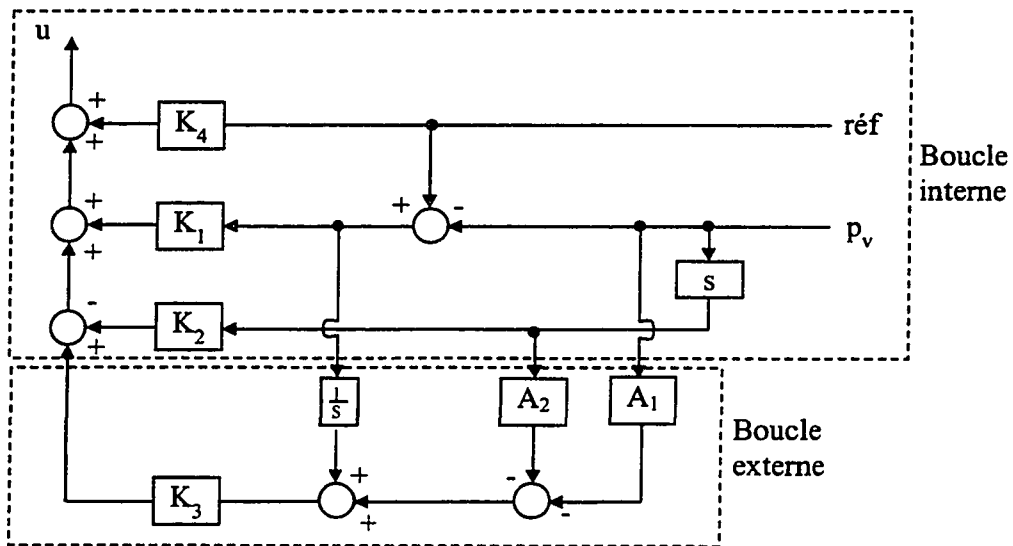


Figure 1.1: Structure du compensateur PID Dual Loop

La relation entrée/sortie du compensateur est donnée par l'expression:

$$u(t) = K_1(\text{réf} - p_v) - K_2 \dot{p}_v + K_3 \left[\int_0^t (\text{réf} - p_v(x)) dx - A_1 p_v - A_2 \dot{p}_v \right] + K_4 \text{réf} \quad (1.1)$$

où

K_1 représente le gain proportionnel,

K_2 représente le gain dérivatif,

K_3 représente le gain intégral,

K_4 représente le gain anticipateur,

A_1 représente le gain de la boucle externe sur la variable du procédé,

A_2 représente le gain de la boucle externe sur la dérivée de la variable du procédé,

p_v variable du procédé,

réf point de consigne (set point).

1.3 Deux propriétés caractéristiques du PID Dual Loop

L'intérêt du Dual Loop réside dans la possibilité de régler indépendamment la dynamique et la robustesse du système asservi. Pour fins d'illustrations, considérons le système à la figure 1.2: soit la fonction de transfert du processus G , du type:

$$G_1(s) = \frac{K_M}{1 + s\tau_p} \quad \text{ou du type} \quad G_2(s) = \frac{K_M}{(1 + s\tau_p)s}$$

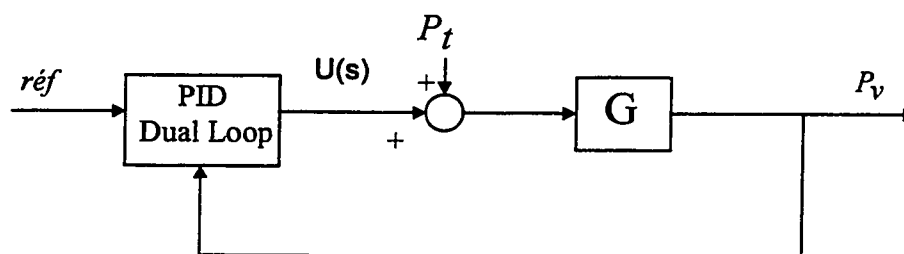


Figure 1.2: Schéma-bloc d'un asservissement

Les fonctions de transfert en boucle fermée sont données par:

- Pour $G_1(s)$

$$F_1(s) = \frac{P_v(s)}{réf} = \frac{(K_4 + K_1) K_M s + K_3 K_M}{(\tau_p + (K_2 + A_2 K_3) K_M) s^2 + (1 + (K_1 + A_1 K_3) K_M) s + K_3 K_M} \quad (1.2)$$

$$F_2(s) = \frac{P_v(s)}{P_i(s)} = \frac{K_M s}{(\tau_p + (K_2 + A_2 K_3) K_M) s^2 + (1 + (K_1 + A_1 K_3) K_M) s + K_3 K_M} \quad (1.3)$$

À partir des expressions (1.2) et (1.3), on peut déduire deux propriétés: la première est la satisfaction des contraintes de performances dynamiques (commande/sortie), la deuxième consiste à améliorer la robustesse (De Santis, 1994).

Pour illustrer la première propriété, les gains de la boucle externe du compensateur PID Dual Loop sont nuls; pour la boucle interne, on pose $K_2 = 0$, $K_4 = \frac{1}{K_M}$ et on doit choisir d'une façon adéquate le gain K_1 ; le système étant soumis à l'action d'un compensateur proportionnel, alors on obtient en boucle fermée les fonctions suivantes:

$$F_{10}(s) = \frac{1}{1 + \tau_{BF} s} \quad (1.4)$$

où

$$\tau_{BF} = \frac{\tau_p}{1 + K_1 K_M}$$

$$F_{20}(s) = \frac{K_M}{(1 + K_1 K_M)(1 + \tau_{BF} s)} \quad (1.5)$$

La deuxième propriété consiste à améliorer la robustesse. Alors, on doit conserver les mêmes gains K_1 , K_2 et K_4 de la première propriété et ajouter les gains de la boucle externe en prenant $A_1 = \tau_{BF}$, $A_2 = 0$ et $K_3 > 0$. On démontre alors que:

$$F_1(s) = \frac{(K_4 + K_1)K_M s + K_3 K_M}{((1 + K_1 K_M)s + K_3 K_M)(1 + \tau_{BF} s)} \quad (1.6)$$

donc
$$F_1(s) = \frac{1}{1 + \tau_{BF} s} = F_{10}(s) \quad (1.7)$$

$$F_2(s) = \frac{K_M s}{(1 + K_1 K_M)(1 + \tau_{BF} s) \left(s + \frac{K_3 K_M}{1 + K_1 K_M} \right)} \quad (1.8)$$

- Pour $G_2(s)$

On obtient les fonctions de transfert suivantes:

$$F_1(s) = \frac{\left(1 + \frac{K_4}{K_1} \right) s + \frac{K_3}{K_1}}{\left(1 + \frac{(1 + K_2 K_M)}{K_1 K_M} s + \frac{\tau_p}{K_1 K_M} s^2 \right) s + \left(\frac{1 + A_1 s + A_2 s^2}{K_1} \right) K_3} \quad (1.9)$$

$$F_2(s) = \frac{\frac{s}{K_1}}{\left(1 + \frac{(1 + K_2 K_M)}{K_1 K_M} s + \frac{\tau_p}{K_1 K_M} s^2\right) s + \frac{(1 + A_1 s + A_2 s^2) K_3}{K_1}} \quad (1.10)$$

Comme dans le cas précédent, en prenant $A_1 = 0$, $A_2 = 0$, $K_3 = 0$ et $K_4 = 0$, on obtient la fonction de transfert:

$$F_{10}(s) = \frac{1}{\frac{1}{\omega_n^2} s^2 + \frac{2\xi}{\omega_n} s + 1} \quad (1.11)$$

$$F_{20}(s) = \frac{1}{\left(\frac{1}{\omega_n^2} s^2 + \frac{2\xi}{\omega_n} s + 1\right) K_1} \quad (1.12)$$

où ω_n est la fréquence naturelle et ξ est l'amortissement de la fonction de deuxième ordre.

Les paramètres intervenant dans les équations (1.11) et (1.12) satisfont les relations suivantes:

$$\frac{1}{\omega_n^2} = \frac{\tau_p}{K_1 K_M} \quad (1.13)$$

$$\frac{2\xi}{\omega_n} = \frac{1 + K_2 K_M}{K_1 K_M} \quad (1.14)$$

Pour démontrer la deuxième propriété, on suppose que K_1 et K_2 ont bien été choisis à l'étape précédente et $K_4 = 0$. On prend les gains de la boucle externe de la façon suivante:

$$A_1 = \frac{2\xi}{\omega_n} = \frac{1 + K_2 K_M}{K_1 K_M} \quad (1.15)$$

$$A_2 = \frac{1}{\omega_n^2} = \frac{\tau_p}{K_1 K_M} \quad (1.16)$$

Alors, la fonction $F_1(s)$ devient:

$$F_1(s) = \frac{\left(s + \frac{K_3}{K_1}\right)}{\left(\frac{1}{\omega_n^2} s^2 + \frac{2\xi}{\omega_n} s + 1\right) \left(s + \frac{K_3}{K_1}\right)} \quad (1.17)$$

donc

$$F_1(s) = \frac{1}{\left(\frac{1}{\omega_n^2} s^2 + \frac{2\xi}{\omega_n} s + 1\right)} = F_{10}(s) \quad (1.18)$$

$$F_2(s) = \frac{s}{\left(\frac{1}{\omega_n^2} s^2 + \frac{2\xi}{\omega_n} s + 1 \right) K_1 \left(s + \frac{K_3}{K_1} \right)} \quad (1.19)$$

Les deux propriétés du compensateur PID Dual Loop (De Santis, 1994) peuvent se résumer de la manière suivante:

Si P_i est une perturbation ajoutée sur la commande et si $F_{10}(s) = \frac{P_v(s)}{\text{réf}}$ et

$F_{20}(s) = \frac{P_v(s)}{P_i(s)}$ sont les fonctions de transfert lorsqu'on n'utilise pas la boucle externe

(A_1 , A_2 et K_3 nuls) alors, en conservant les mêmes gains K_1 , K_2 , K_4 et en ajoutant les gains A_1 , A_2 et K_3 adéquats, on obtient:

$$F_1(s) = F_{10}(s) \text{ (fonction de transfert entrée/sortie conservée)} \quad (1.20)$$

$$F_2(s) = \Gamma(s) F_{20}(s) \text{ avec } \left\{ \begin{array}{l} \Gamma(s) = \frac{s}{K_3 K_M} \text{ pour } G_1(s), \quad K_4 = \frac{1}{K_M} \\ s + \frac{K_3}{K_1} \\ 1 + K_1 K_M \end{array} \right. \text{ et } \left\{ \begin{array}{l} \Gamma(s) = \frac{s}{K_3} \text{ pour } G_2(s), \quad K_4 = 0 \\ s + \frac{K_3}{K_1} \end{array} \right. \quad (1.21)$$

donc, en théorie, on peut obtenir le degré de robustesse souhaitée face à la perturbation tout simplement en augmentant K_3 .

Ces propriétés suggèrent l'adoption d'une procédure de réglage du compensateur PID Dual Loop du type suivant:

- réglage de K_1 , K_2 et K_4 de façon à obtenir une dynamique satisfaisante,
- introduction des gains A_1 , A_2 et augmentation de K_3 jusqu'à obtenir la robustesse souhaitée. Voir l'organigramme de la solution du problème en utilisant le PID Dual Loop à l'annexe 8 et la procédure de réglage de leurs paramètres à l'annexe 9.

Il est ensuite important de vérifier si la dynamique obtenue avec les seuls gains K_1 , K_2 et K_4 est maintenue lorsqu'on introduit A_1 , A_2 et K_3 (cf équation (1.20)) et dans quelle mesure on peut augmenter K_3 sans détériorer la réponse.

1.4 Procédure de réglage automatique des gains

Le module d'adaptation, qu'on a développé pour obtenir les gains automatiques du compensateur PID Dual Loop a la structure de la figure 1.3. Cette structure a deux entrées et deux sorties, et implique l'utilisation de deux compensateurs PID Dual Loop.

Les gains de ces compensateurs seront indiqués avec les symboles K_{11} , K_{12} , K_{13} , K_{14} , A_{11} , A_{12} (cas monovariables, figure 1.3) lorsqu'on ne considère qu'une paire d'entrée-sortie; avec les symboles K_{21} , K_{22} , K_{23} , K_{24} , A_{21} , A_{22} , lorsqu'on considère simultanément les deux paires (cas multivariable, figure 1.5). Dans le cas du procédé représenté par le banc d'essai que nous avons mis en place et qui sera décrit dans le chapitre suivant, la stratégie de réglage est décrite dans les paragraphes suivants.

1.4.1 Cas monovariante

Le schéma-bloc de l'autoréglage de l'asservissement est représenté à la figure 1.3.

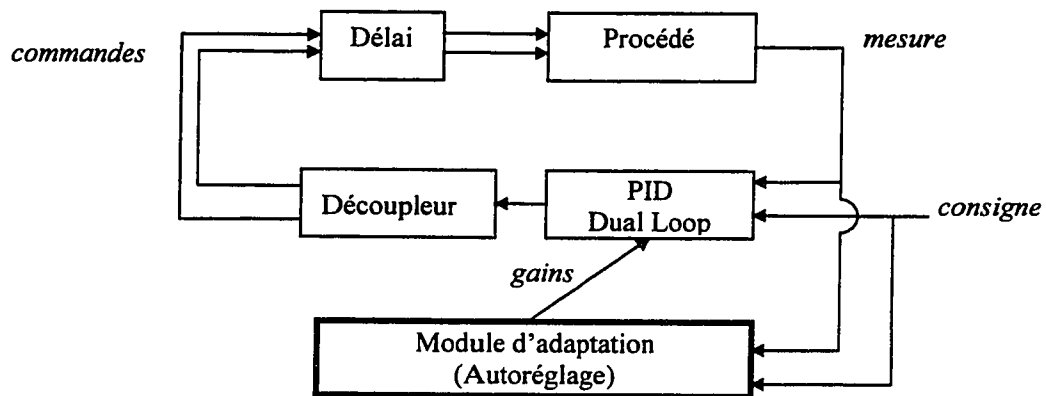


Figure 1.3: Schéma-bloc de l'autoréglage (monovariante)

Le réglage automatique se compose de deux étapes détaillées dans les deux paragraphes suivants.

1.4.2 Première étape du réglage

La première étape consiste à trouver un réglage satisfaisant pour K_1 et K_2 . Pendant cette étape les seuls gains non nuls sont K_1 et K_2 (appelés k_{10} et k_{20} avant une itération et k_1 et k_2 après cette itération).

Au début du programme, l'utilisateur indique l'amplitude *ampl* et le temps *tpd* maximaux de premier dépassement qu'il aimerait voir respecter à l'issue de la première étape, c'est-à-dire lorsqu'on règle le compensateur avec les deux gains K_1 et K_2 trouvés (A_1 , A_2 et K_3 nuls).

La première étape s'arrête lorsque ces critères sont satisfaits ou lorsque le programme détecte des oscillations excessives dans la réponse.

Pendant cette étape, le programme procède à plusieurs essais en modifiant à chaque fois les réglages de K_1 et K_2 selon la stratégie décrite ci-dessous.

Quand $K_3=0$, $K_4=0$, $A_1=0$, $A_2=0$ et $P_7=0$, la figure 1.3 est équivalente à la figure 1.4:

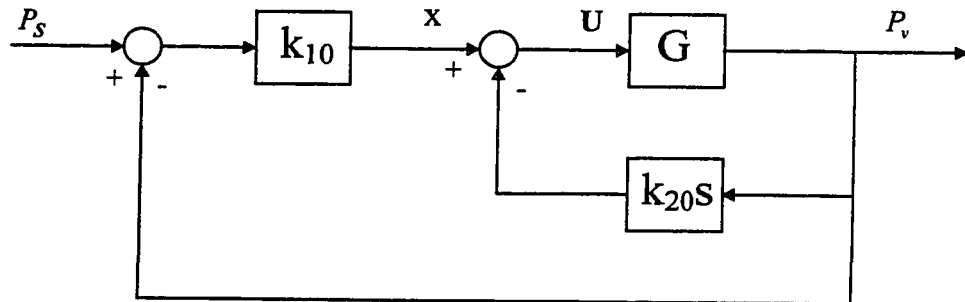


Figure 1.4: Schéma-bloc de l'asservissement pendant la première étape de réglage

On remarque que cette structure correspond à la loi de commande d'un compensateur PD quand elle est implantée sous la forme série avec une action dérivée uniquement sur la mesure pour éviter les écarts brusques de commande.

Cette structure permet d'établir les équations suivantes:

$$\frac{P_v}{x} = \frac{G}{1 + Gk_{20}s} \quad \text{et} \quad x = (P_s - P_v)k_{10}$$

En combinant ces équations, la fonction de transfert en chaîne fermée devient:

$$\frac{P_v}{P_s} = \frac{Gk_{10}}{1 + Gk_{10} + Gk_{20}s}$$

Supposons que le processus peut être modélisé par une fonction de transfert du second ordre:

$$G = \frac{K}{a_2 s^2 + a_1 s + 1} \quad (1.22)$$

alors

$$\frac{P_v}{P_s} = \frac{1}{\frac{a_2 s^2}{Kk_{10}} + \left(\frac{a_1 + Kk_{20}}{Kk_{10}} \right) s + \frac{1 + Kk_{10}}{Kk_{10}}}$$

En considérant que

$$Kk_{10} + 1 \approx Kk_{10} \quad (1.23)$$

il s'ensuit que la fonction de transfert du système en boucle fermée est

$$\frac{P_v}{P_s} \approx \frac{1}{\alpha_{20} s^2 + \alpha_{10} s + 1} = \frac{1}{\frac{1}{\omega_{n0}^2} s^2 + \frac{2\xi}{\omega_{n0}} s + 1}$$

avec:

$$\alpha_{20} = \frac{a_2}{Kk_{10}} \quad (1.24)$$

$$\alpha_{10} = \frac{a_1 + Kk_{20}}{Kk_{10}} \quad (1.25)$$

$$\omega_{n0} = \frac{1}{\sqrt{\alpha_{20}}} \quad (1.26)$$

$$\xi_0 = \frac{\alpha_{10} \omega_{n0}}{2} \quad (1.27)$$

Appelons ω_n , ξ , k_1 et k_2 les nouvelles valeurs des variables ω_{n0} , ξ_0 , k_{10} et k_{20} après une première itération.

Supposons que l'on désire calculer k_1 et k_2 de façon à obtenir:

$$\omega_n = \mu \omega_{n0} \quad (1.28)$$

$$\xi = \mu_k \xi_0 \quad (1.29)$$

D'après (1.26), pour satisfaire l'équation (1.28), il est nécessaire de diviser α_{20} par μ^2 ,

$$\text{c'est-à-dire : } \alpha_2 = \frac{\alpha_{20}}{\mu^2} \quad \text{ou encore} \quad \frac{a_2}{Kk_1} = \frac{1}{\mu^2} \frac{a_2}{Kk_{10}} \quad (\text{cf équation (1.24)}).$$

Ce qui implique

$$k_1 = \mu^2 k_{10} = \frac{\alpha_{20}}{\alpha_2} k_{10} \quad (1.30)$$

D'autre part, d'après (1.27), la condition (1.29) est équivalente à la condition

$$\alpha_1 = \frac{2\mu_k \xi}{\omega_n} \quad (1.31)$$

soit encore (d'après (1.25)): $\frac{a_1 + Kk_2}{Kk_1} = \frac{2\mu_k \xi}{\omega_n}$ d'où $k_2 = \frac{2\mu_k \xi}{\omega_n} k_1 - \frac{a_1}{K}$

ce que l'on peut aussi écrire: $k_2 = \left(\frac{Kk_{10}\alpha_{10}}{K} - k_{10}\alpha_{10} \right) - \frac{a_1}{K} + \frac{2\mu_k \xi}{\omega_n} k_1$

$$k_2 = \frac{Kk_{10}\alpha_{10} - a_1}{K} + \frac{\alpha_{20}}{\alpha_2} k_{10} \frac{2\mu_k \xi}{\omega_n} - k_{10}\alpha_{10}$$

Finalement, d'après (1.24) et (1.31), on obtient:

$$k_2 = k_{20} + k_1\alpha_1 - k_{10}\alpha_{10} \quad (1.32)$$

En supposant que les hypothèses (1.22) et (1.23) sont valables, on dispose donc des lois d'évolution des gains K_1 et K_2 (équations (1.30) et (1.32)).

En pratique, la première étape utilise seulement un gain proportionnel (initialisé par l'utilisateur) et après chaque test, le programme de réglage automatique analyse la réponse obtenue.

Remarque:

Si cette réponse présente des oscillations importantes, la première étape s'arrête et les gains K_1 et K_2 retenus pour la seconde étape (réglage de K_3) sont ceux de l'avant-dernier test effectué.

Sinon, on calcule le temps t_{fpa} et l'amplitude *overshoot* du premier dépassement ainsi que la pulsation naturelle et l'amortissement de l'approximation du second ordre de la réponse obtenue.

- si les critères concernant le premier dépassement sont satisfaits, le programme passe à la seconde étape de réglage et les gains K_1 et K_2 retenus sont ceux du dernier test;
- sinon le programme reste dans la première étape de réglage et calcule de nouveaux gains K_1 et K_2 de façon à avoir au test suivant:

$$\omega_n = \mu \omega_{n0} \quad \text{si} \quad t_{fpa} > t_{pd}$$

$$\omega_n = \omega_{n0} \quad \text{si} \quad t_{fpa} < t_{pd} \text{ (ce qui provoque } k_1 = k_{10})$$

$$\xi = \mu_k \xi_0 \quad \text{si} \quad overshoot > ampl$$

$$\xi = \xi_0 \quad \text{si} \quad overshoot < ampl$$

1.4.3 Seconde étape du réglage

Cette seconde étape consiste à trouver la plus grande valeur acceptable pour K_3 . On attribue maintenant à A1 et A2 la valeur des paramètres α_1 et α_2 correspondant à la meilleure réponse obtenue pendant la première étape du réglage. Pendant toute cette étape les paramètres K1, K2, A1 et A2 sont constants.

Cette démarche s'inspire du résultat suivant:

On a approximé la réponse du système asservi par un compensateur proportionnel dérivé (K_1 et K_2).

Or, d'après De Santis (1994), dans le cas d'un système de fonction de transfert

$$G_2(s) = \frac{K_M}{(1 + s\tau_p)s}, \text{ la fonction de transfert du système asservi par un compensateur}$$

proportionnel dérivé (correspondant à la figure 1.4) est une fonction de deuxième ordre:

$$F_{10}(s) = \frac{P_v(s)}{P_i(s)} = \frac{1}{1 + 2\frac{\xi}{\omega_n}s + \frac{1}{\omega_n^2}s^2} \quad (1.33)$$

et la fonction de transfert perturbation/sortie (perturbation ajoutée sur la commande) est

$$F_{20}(s) = \frac{P_v(s)}{P_i(s)} \quad (1.34)$$

Si l'on pose $\alpha_1 = 2\frac{\xi}{\omega_n}$ et $\alpha_2 = \frac{1}{\omega_n^2}$ alors

- la fonction de transfert du système asservi par le PID Dual Loop est

$$F_1(s) = F_{10}(s)$$

- la fonction de transfert perturbation/sortie devient

$$F_2(s) = \frac{s}{s + \frac{K_3}{K_1}} F_{20}(s) \quad (1.35)$$

Dans la deuxième étape du réglage, on cherche à augmenter K_3 autant que possible afin d'améliorer la robustesse du système.

1.4.4 Cas multivariable

Le schéma-bloc de la figure 1.3 est ici transformé de la façon suivante:

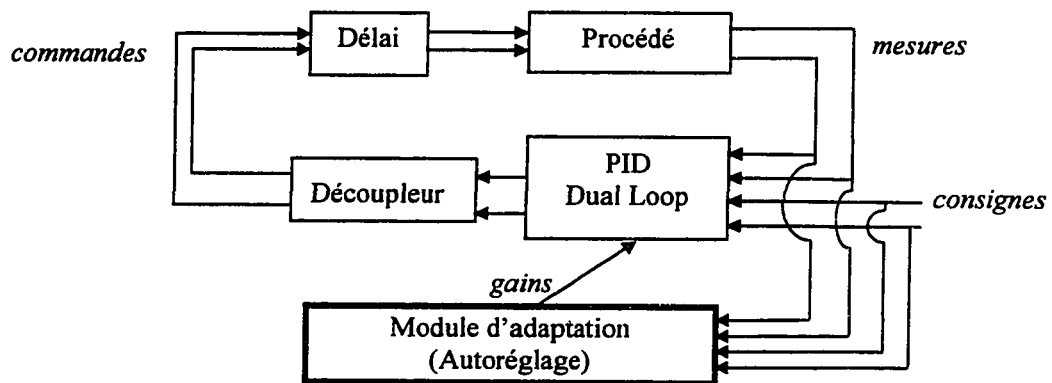


Figure 1.5: Schéma-bloc de l'autoréglage (multivariable)

Le principe de réglage est le même que dans le cas monovariable sauf que:

- la fin de la première étape est conditionnée par 6 critères au lieu de 3 (critères concernant les oscillations ainsi que l'amplitude et le temps de premier dépassement de la réponse);
- dans la seconde étape du réglage, on commence par ajouter progressivement un nombre égal d'incréments incK_{13} et incK_{23} (la valeur de chacun ayant été déterminée au préalable) à K_{13} et K_{23} jusqu'au moment où la sortie de l'un des deux est atteinte puis on essaie d'augmenter indépendamment K_{13} et K_{23} jusqu'à l'invalidation des critères d'oscillations et d'amplitudes de premier dépassement.

1.5 Organigramme de l'algorithme de réglage automatique

L'organigramme correspondant aux méthodes décrites à la section précédente est dans l'annexe 5.

L'ensemble compensateurs-modules d'adaptation est un programme développé avec le progiciel Matlab[®]. Ce programme (pintpri2.m) fait appel à plusieurs sous-programmes (pintauto.m, raz.m, reglak3.m, newgain.m, ordre.m, rech_osc.m) dont le listing est présenté en annexe 10.

CHAPITRE II

IMPLANTATION D'UN BANC D'ESSAI

2.1 Introduction

L'objectif de ce chapitre est de présenter un banc d'alimentation en eau, l'instrumentation associée et le système de contrôle qu'on a mis en place afin de nous doter d'un banc d'essai émulant un procédé industriel typique.

2.2 Description générale du banc d'essai

Avec référence à la figure 2.1, les principales composantes du banc d'essai sont représentées par: un réservoir, deux contrôleurs de débit, un ensemble d'instruments de mesure et de contrôle. Une description détaillée du banc d'essai peut être retrouvée dans la référence Cornieles et Bougeret (1997). Dans ce qui suit, nous en présenterons les éléments principaux.

2.2.1 Réservoir

Le réservoir a une configuration géométrique cylindrique. Il est alimenté par deux arrivées d'eau: l'une pour l'eau chaude et l'autre pour l'eau froide. Les débits livrés par ces deux entrées sont contrôlés par deux vannes indépendantes. C'est en agissant sur

ces deux éléments du système que l'on va réaliser la régulation du niveau et de la température. Le réservoir dispose d'un robinet permettant de régler manuellement le débit de sortie. On dispose de deux thermocouples qui permettent de mesurer les températures des arrivées d'eau froide et d'eau chaude. Un troisième thermocouple est fixé sur le robinet de sortie. Il donne la température du mélange contenu à l'intérieur du réservoir, donc la température que l'on désire réguler. Le système est également équipé d'un capteur de pression qui est fixé dans la partie basse du réservoir. Il donne une information sur le niveau d'eau que l'on veut réguler. Les descriptions précédentes complètent le schéma de la figure 2.1 montrant les deux boucles d'asservissement. F2 et F3 sont des robinets normalement fermés mais utilisés pour accélérer le remplissage manuel du réservoir (voir photo en annexe 1). La liste des composantes physiques utilisées et les gains des capteurs se trouvent en annexe 2. Les différentes interfaces de la figure 2.1 se trouvent en annexe 4.

La cuve ayant une capacité relativement importante, la température de l'eau varie très lentement et n'est pas homogène à l'intérieur du réservoir. Les débits d'eau froide et d'eau chaude arrivant en surface mettent un certain temps pour se propager jusqu'à la sortie. L'asservissement de température est donc très délicat à effectuer avec précision. Le réservoir dispose d'un agitateur que l'on n'a pas utilisé pour les expériences présentées dans ce rapport à cause du bruit électromagnétique et des fluctuations de niveau qu'il introduit.

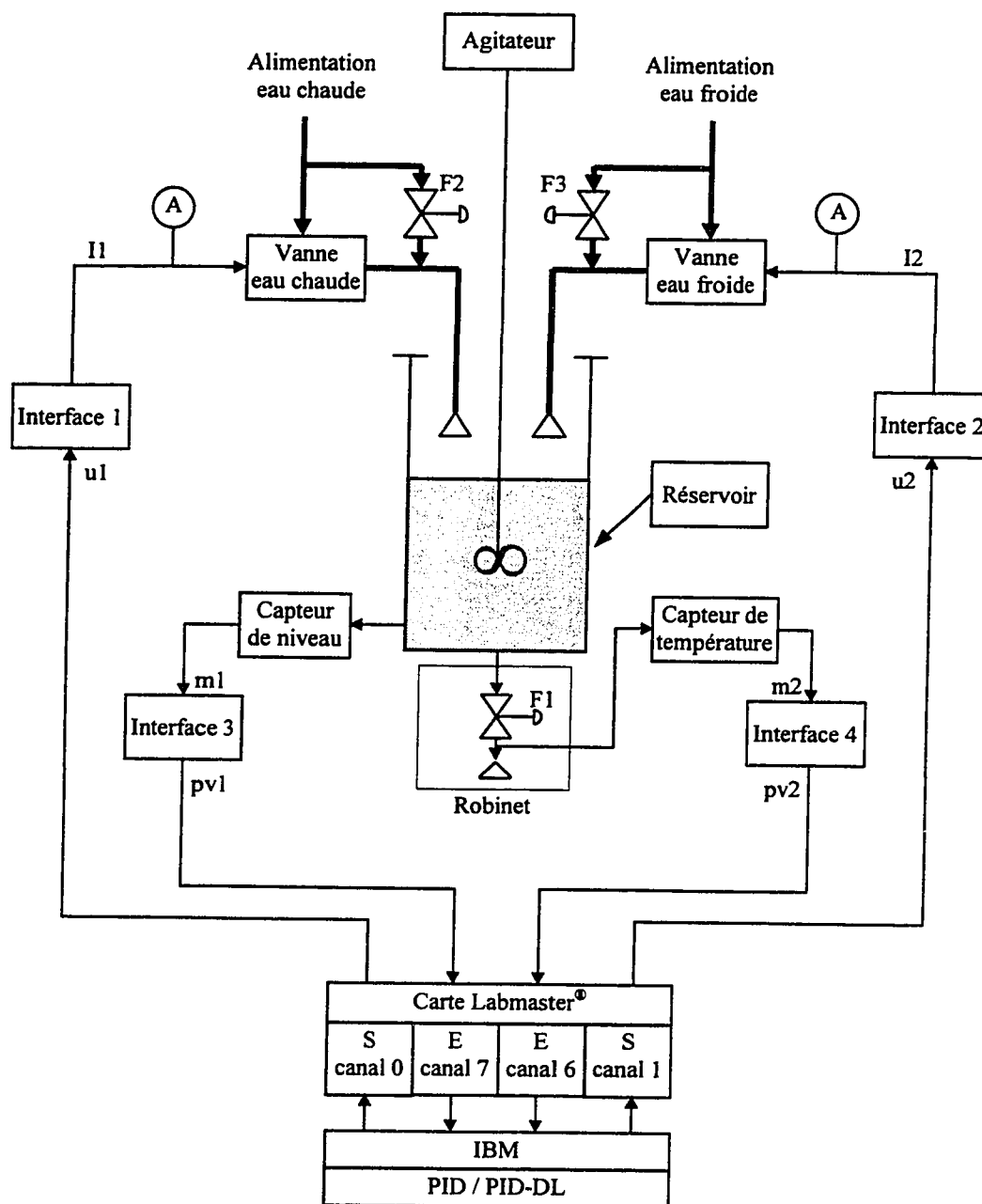


Figure 2.1: Représentation du système complet

2.2.2 Interaction entre le système et le compensateur

Le compensateur est constitué d'un PID classique/PID Dual Loop et est réalisé à l'aide d'un PC équipé d'un microprocesseur Pentium® 166MHz. Pour le traitement des signaux d'entrée et de sortie du système, le micro-ordinateur utilise une carte entrée/sortie Labmaster®. Cette carte dispose de deux canaux numérique/analogique et seize canaux analogique/numérique.

La carte Labmaster® fournit au compensateur deux signaux: l'un représentatif du niveau d'eau contenue dans le réservoir, l'autre de la température du liquide. On appellera pv1 la mesure du niveau et pv2 la mesure de la température. À la figure 2.2, on trouve l'organigramme d'interactions entre le système et le compensateur.

Pour agir sur le système, le compensateur utilise deux canaux numérique/analogique de la carte Labmaster®. L'un permet de commander la vanne d'eau froide, l'autre la vanne d'eau chaude à travers un découpleur qui sera présenté au chapitre III. On appellera u1 la commande d'eau chaude et u2 la commande d'eau froide. Les interactions entre compensateur et système sont ainsi parfaitement définies.

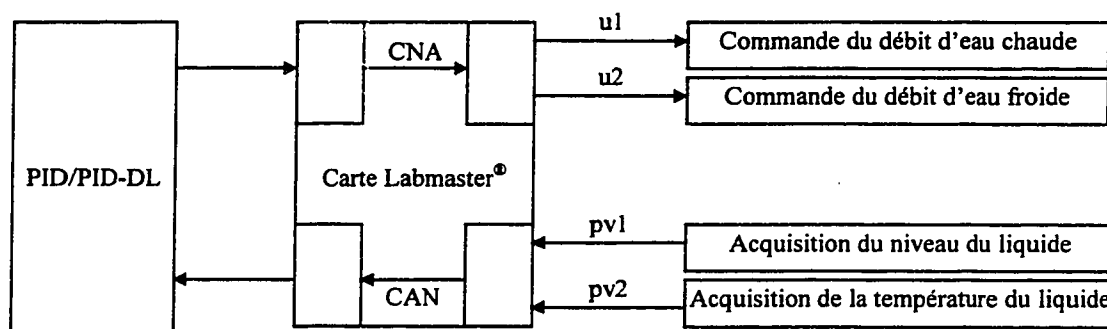


Figure 2.2: Interactions entre système et compensateur

2.2.3 Commande des vannes d'eau chaude et d'eau froide

Les vannes d'alimentation en eau sont des vannes pneumatiques dont la commande est un signal de pression variant de 3 à 15psi.

La caractéristique débit/tension de ces vannes est considérée linéaire dans le domaine où l'on travaille. Les coefficients de ces vannes sont:

$K_{v1} = 0.24$ litre/secondes x volts (vanne d'eau froide) et $K_{v2} = 0.126$ litre/secondes x volts (vanne d'eau chaude).

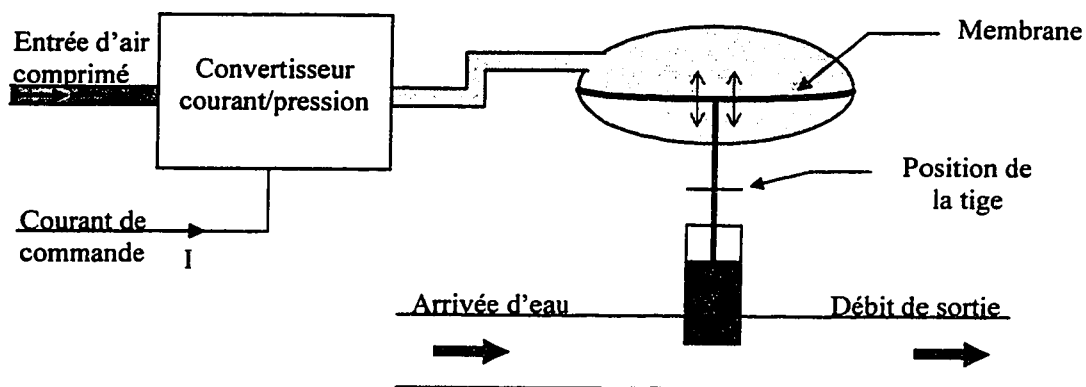


Figure 2.3: Schématisation d'une vanne.

Le signal appliqué à la vanne provient d'un convertisseur courant électrique/pression pneumatique. Une variation de cette pression déforme la membrane de caoutchouc. La position de la membrane est donc définie par la pression d'entrée. La membrane est liée par une tige métallique à un "bouchon" qui va plus ou moins obstruer la conduite de liquide selon sa position. Cette variation du coefficient de perte de charge de la conduite provoque une variation du débit. Ainsi le débit de la vanne est fonction du courant de commande.

2.2.4 Conditions initiales du système

Pour que le niveau puisse être régulé, le compensateur doit pouvoir augmenter ou diminuer le volume de liquide. Pour que le niveau puisse diminuer, il est nécessaire que le débit de sortie soit non nul à l'état de repos. Le réservoir dispose donc d'un robinet

de sortie qui sera ouvert de la même façon pour toutes les expériences.

On désire avoir un système stable au point de repos. La somme des débits d'entrée doit donc compenser le débit de sortie lorsque les commandes d'eau chaude u_1 et d'eau froide u_2 sont nulles. Les relations entre les commandes u_1 , u_2 et les signaux V_1 , V_2 appliquées aux vannes doivent donc être de la forme:

$$V_1 = \text{offset1} + u_1 \quad (2.1)$$

$$V_2 = \text{offset2} + u_2 \quad (2.2)$$

Pour avoir ce type de comportement, on utilise des amplificateurs opérationnels dans un montage de type sommateur.

Pour mettre le système au point de repos, on fixe $u_1=u_2=0$ et on ajuste les potentiomètres de manière à avoir une égalité entre les débits d'entrée et de sortie. Les deux débits initiaux maintiennent le niveau stable et fixent la température du liquide lorsque le système est au point de repos. Cet équilibrage n'est pas facile car les tensions V_1 et V_2 doivent demeurer entre 0 et 7V (pour conserver la linéarité) et car les potentiomètres analogiques utilisés sont très sensibles à cause de leur gain élevé (15). De plus, la pression d'eau livrée par le réseau n'est pas constante.

La stabilité du niveau dépend notamment de la finesse de l'équilibrage entre les débits d'entrée et de sortie. Pour visualiser le niveau, on utilise un montage de type sommateur représenté à l'annexe 4. Le montage permet également de mettre la tension à 0 pour un certain niveau (généralement 0.5m).

2.2.5 Modèle du système

Des tests pour l'identification du procédé (Cornieles et Bougeret, 1997) ont permis de conclure que la fonction de transfert entre la tension de contrôle U_1 et le capteur de niveau peut être approximée avec une fonction du premier ordre. Il en est de même pour la température. Chaque fonction de transfert est caractérisée par une constante de temps et par un gain statique. La fonction de transfert du niveau en boucle ouverte est

$$\frac{8.9}{1 + 332 S}.$$

Pour déterminer les gains du PID au chapitre III avec les techniques classiques de réglage, on ajoute un retard 30s à cette fonction de transfert, c'est-à-dire qu'on utilise la

fonction de transfert suivante: $\frac{8.9 e^{-30 s}}{1 + 332 S}.$

En effet, lorsqu'on comparera les performances des différents réglages du PID, on introduira un retard de 30s sur la commande car la constante de temps du niveau en

boucle fermée (avec $H=1$) sans compensateur est $\frac{332}{8.9+1} \approx 33.5$ s. Un retard de 30s est

donc suffisant pour perturber le système de façon significative.

En ce qui concerne la fonction de transfert entre la tension de contrôle de U2 et la sortie de capteur de la température trouvée dans les travaux précédents (Cornieles et Bougeret,

1997) est $\frac{1.26 e^{-20 s}}{1 + 305 S}$.

2.2.6 Logiciel d'asservissement

La structure du logiciel est constituée de trois parties: la première partie concerne l'interaction entre le logiciel et l'utilisateur. Elle permet le choix de l'asservissement, la modification des gains du compensateur, la possibilité d'ajouter un retard et/ou une perturbation et le lancement de l'expérience. La deuxième partie concerne l'interaction entre le banc d'essai, la carte d'acquisition des données et le compensateur. Cette partie est très importante car lorsque l'expérience est terminée le logiciel donne l'évolution des variables du procédé et de la commande de chaque boucle en fonction du temps. La troisième partie est relative à la sauvegarde des données d'expériences sous forme de fichier compatible avec le langage Matlab®.

La figure 2.4 montre l'organigramme du logiciel de l'asservissement du niveau et de la température. Le logiciel utilisé est `pint_eds.m` et a été écrit avec le langage commercial Matlab®. Le logiciel a été implanté sur le banc d'essai pour l'asservissement du niveau et de la température en temps réel. Le programme complet se trouve à l'annexe 8.

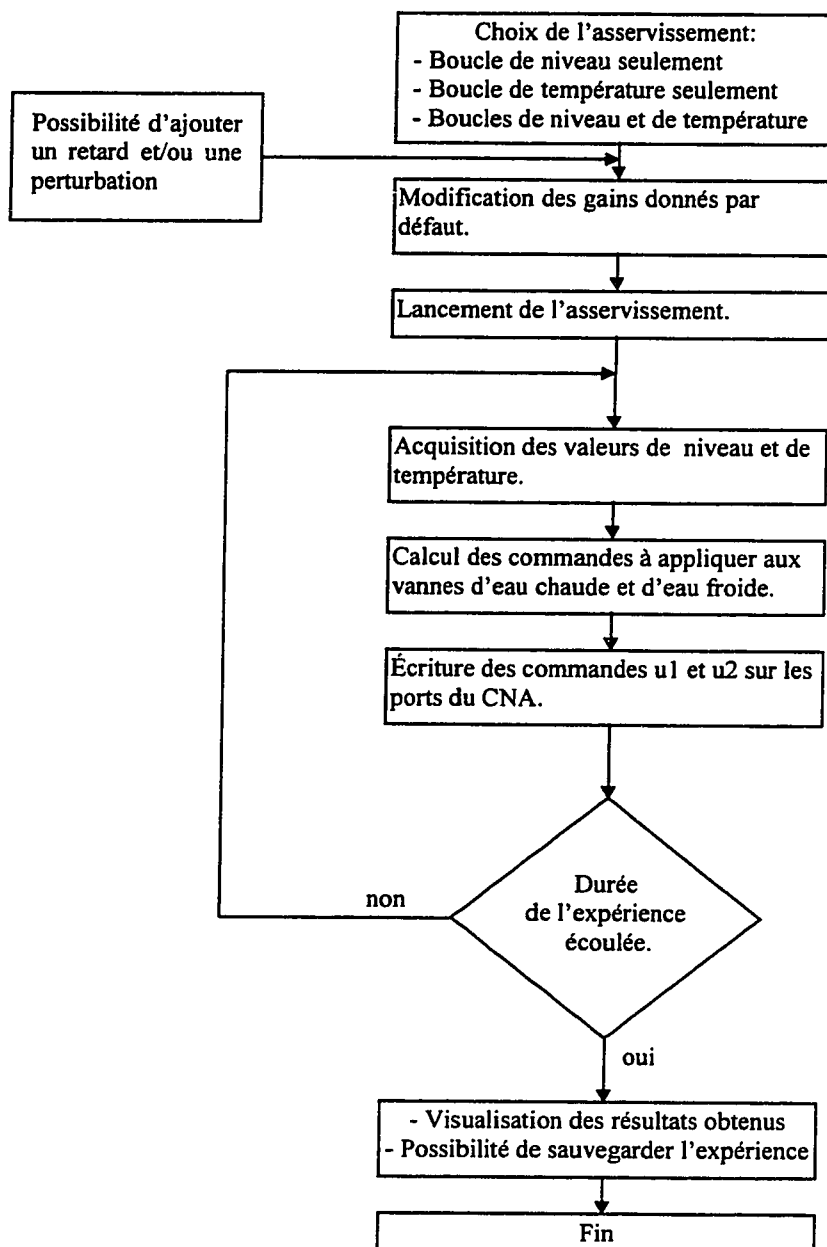


Figure 2.4: Organigramme du logiciel d'asservissement.

CHAPITRE III

ÉTUDE COMPARATIVE EXPÉRIMENTALE

DES COMPENSATEURS PID ET PID DUAL LOOP

3.1 Introduction

L'objectif de ce chapitre est de présenter les résultats de nos études concernant une comparaison expérimentale de la procédure de réglage du compensateur PID classique (présentée à l'annexe 7) et de la procédure de réglage du compensateur PID Dual Loop pour l'asservissement du niveau et de la température du banc d'essai (présentée aux chapitres précédents).

3.2 Différentes techniques de réglage des compensateurs PI classique et PID Dual Loop

3.2.1 Réglage du compensateur PI classique

Les fonctions de transfert du niveau et de la température étant représentables par un système de premier ordre suivi par un retard, nous suivrons des techniques standards (reprises à l'annexe 7) actuellement en vigueur dans l'industrie et nous n'utiliserons pas d'actions dérivées dans la structure du PID classique.

La fonction de transfert du compensateur PI classique utilisée dans toutes les références est de la forme:

$$U(S) = K1(réf - P_v(s)) + K3 \left(\frac{(réf - P_v(s))}{s} \right) \quad (3.1)$$

3.2.2 Réglage du compensateur PID Dual Loop

Les gains du compensateur PID Dual Loop (De Santis, 1994) sont déterminés en exploitant la structure du PID Dual Loop. La réponse temporelle entrée-sortie du système asservi est obtenue avec un compensateur PD classique (réglé de façon à avoir un régime transitoire assez satisfaisant). Voir les organigrammes des annexes 8 et 9. La démarche présentée au chapitre I a permis de développer un logiciel permettant d'obtenir les gains du compensateur PID Dual Loop. Ce logiciel permet d'obtenir d'une façon claire l'identification de l'approximation du second ordre décrite au paragraphe 1.3. L'organigramme du logiciel se trouve à l'annexe 6.

La procédure de réglage des gains est la suivante:

On enregistre la réponse temporelle du système asservi avec un compensateur PD classique. Le logiciel donne une approximation du second ordre de cette réponse. Il propose également de nouveaux gains K1 et K2 susceptibles d'améliorer le régime transitoire et fournit l'approximation du second ordre qui devrait correspondre à la réponse du système asservi avec les nouveaux gains proposés. La structure du

logiciel se trouve dans l'organigramme de l'annexe 6. Si le régime transitoire obtenu est satisfaisant, on pose $A1=2\xi/\omega_n$ et $A2=1/(\omega_n)^2$ où ξ et ω_n sont le coefficient d'amortissement et la pulsation naturelle de l'approximation du second ordre. On augmente K3 autant que possible (jusqu'à détérioration du régime transitoire) pour réduire la sensibilité aux perturbations.

Pour cette étude expérimentale, on a réglé successivement chacune des boucles de la façon suivante:

On a commencé par asservir la boucle avec un compensateur proportionnel (gain $K11=0.5$ pour le niveau et gain $K21=3$ pour la température). À partir de la réponse obtenue, le programme newgain.m a alors proposé un jeu de gains pour asservir la boucle avec un compensateur PD. On a ensuite testé expérimentalement le compensateur PD et il a donné une réponse satisfaisante. Enfin, le logiciel a permis d'identifier la réponse obtenue avec la réponse d'un système du second ordre et donc d'obtenir A1 et A2.

3.3 Résultats expérimentaux

On présente maintenant les tests réalisés afin de comparer les différentes techniques de réglage décrites au chapitre I. K11, K12, K13, K14, A11, A12 sont les paramètres du compensateur PID Dual Loop pour le niveau et K21, K22, K23, K24, A21, A22 les paramètres du compensateur de température.

Les références sont données en volts. Pour avoir une idée de la variation de niveau, il faut savoir que 2V de variation en sortie du capteur de pression correspondent à 5 centimètres de variation du niveau dans le réservoir. Toutes les expériences sont réalisées avec un niveau de repos de 0.5m (0V), une consigne en niveau de 0.55m (2V), une température de repos de 20°C (2V), une consigne en température de 32°C (5V) et le robinet de sortie ouvert à une position fixée.

3.3.1 Logiciel d'asservissement du niveau et de la température

L'asservissement est réalisé par le logiciel `pint_eds.m` (Cornieles et Bougeret, 1997). Il s'agit d'un programme écrit avec le langage commercial Matlab® et utilisant l'organigramme de la figure 2.4. Le listing du logiciel se trouve à l'annexe 10.

L'asservissement est constitué de deux PID Dual Loop. Chacun de ces compensateurs a la structure suivante:

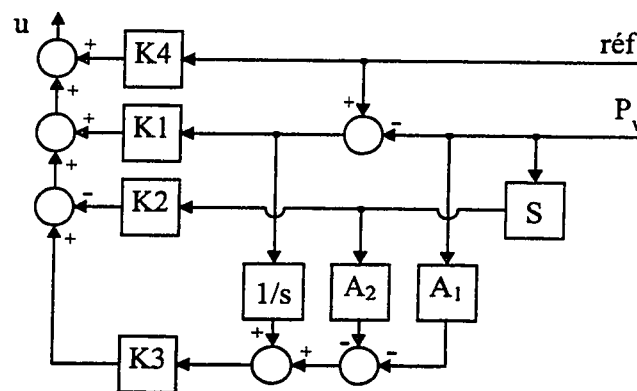


Figure 3.1: Structure du PID Dual Loop.

Ces compensateurs 1 et 2 qui correspondent respectivement à l'asservissement du niveau et de la température peuvent être utilisés seuls ou ensembles en utilisant la structure d'un découpleur (Seborg, Edgar et Mellichamp, 1989; Smith et Carripio, 1997) représentée à la figure 3.2:

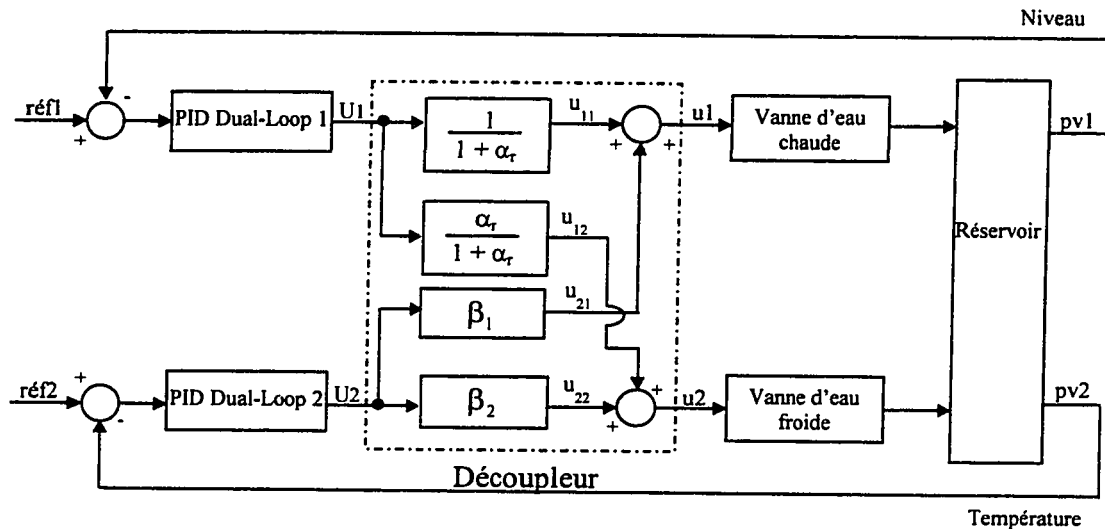


Figure 3.2: Structure générale de l'asservissement.

Les paramètres α_r , β_1 et β_2 sont choisis de façon telle que, dans un régime stationnaire, U1 module les débits d'eau froide et d'eau chaude tout en gardant la température du réservoir constante; U2 module le rapport entre le débit d'eau froide et le débit d'eau chaude tout en gardant le niveau du réservoir constant.

Étant donné que les gains des vannes sont différents ($K_{v1}=0.24$ et $K_{v2}=0.126$), on a posé $\beta_1 = 0.36$ et $\beta_2 = -0.64$. Cela permet de maintenir un niveau constant lorsqu'on fait un asservissement en température seule ($0.36 \cdot 0.24 \cong 0.64 \cdot 0.126$). La valeur de

$\alpha_r = 1.7$ du découpleur a été déterminée de façon expérimentale en fonction de la température de l'eau froide, selon la saison.

L'utilisateur peut modifier pour chaque test:

Les paramètres des PID Dual Loop des deux boucles: K11, K12, K13, K14, A11, A12, K21, K23, K24, A21, A22.

Les consignes réf1 et réf2 du niveau et de la température exprimées en volts.

La période d'échantillonnage du compensateur.

Le choix de perturber le système:

- avec un échelon de 1V ajouté sur la commande de la vanne d'eau froide au milieu de l'expérience (utilisé lors du test de robustesse de l'asservissement de niveau seul et lors du test de robustesse de l'asservissement simultané des deux boucles scalaires);
- avec un échelon de -0.54V ($-0.54 = -1.5 * \beta_1$) sur la commande de la vanne d'eau chaude et un échelon de 0.96V ($0.96 = -1.5 * \beta_2$) sur la commande de la vanne d'eau froide (utilisés lors du test de robustesse de l'asservissement de température seule afin de perturber le moins possible le niveau).

Le choix d'un retard sur la commande.

Le nombre d'échantillons sur lesquels va porter l'expérience. On remarque que le produit de la période d'échantillonnage avec le nombre d'échantillons est égal à la durée de l'expérience.

Un filtre numérique du premier ordre avec une constante de temps de 20s a été ajouté sur la mesure de la température.

Lorsque l'expérience est terminée, le logiciel donne l'évolution des variables du procédé et de la commande de chaque boucle dans le temps. L'utilisateur peut également imprimer et sauvegarder ces mêmes données sous forme de fichier utilisable par Matlab®. On peut également faire la superposition de plusieurs variables et de différentes expériences.

3.4 Tests sur le contrôle du niveau seul

3.4.1 Asservissement du niveau seul

Objectif: Mise en évidence des différences de performances de l'asservissement du niveau en présence d'un retard sur la commande pour plusieurs types de réglage des compensateurs PI et PI Dual Loop.

Modalités: On ouvre le robinet de sortie et on ajuste les potentiomètres $\alpha 1$ et $\alpha 2$ pour équilibrer le débit d'entrée avec le débit de sortie. Lorsque le niveau est stabilisé à 0.5m, on règle le potentiomètre de l'interface 3 de façon à avoir une mesure de 0V. Le système est alors stationnaire et on peut lancer le logiciel d'asservissement. Grâce à ce logiciel, on introduit un retard de 30s sur la commande et on entre le jeu de gains correspondant au type de réglage souhaité. Le temps d'échantillonnage a été déterminé expérimentalement et fixé à 0.5s pour toutes les expériences. Les gains utilisés sont présentés au tableau 3.1 (On a utilisé une seule contrainte pour la valeur λ , $\frac{\lambda}{\alpha} > 1.7$, selon Rivera et Morari, voir annexe 7).

Tableau 3.1: Paramètres des compensateurs PI et PID Dual Loop pour la régulation de niveau

Paramètres	Ziegler-Nichols	Cohen-Coon	Rivera-Morari (IMC), $\lambda=54$	ITAE	PID Dual Loop
K_{11}	1.12	1.13	0.70	0.60	0.605
K_{12}	0	0	0	0	3.793
K_{13}	0.012	0.014	0.002	0.002	0.005
K_{14}	0	0	0	0	0
A_{11}	0	0	0	0	43.23
A_{12}	0	0	0	0	3014

Pour chaque réglage, on fait l'expérience; ensuite, on fait la superposition des courbes obtenues pour comparer et analyser les résultats.

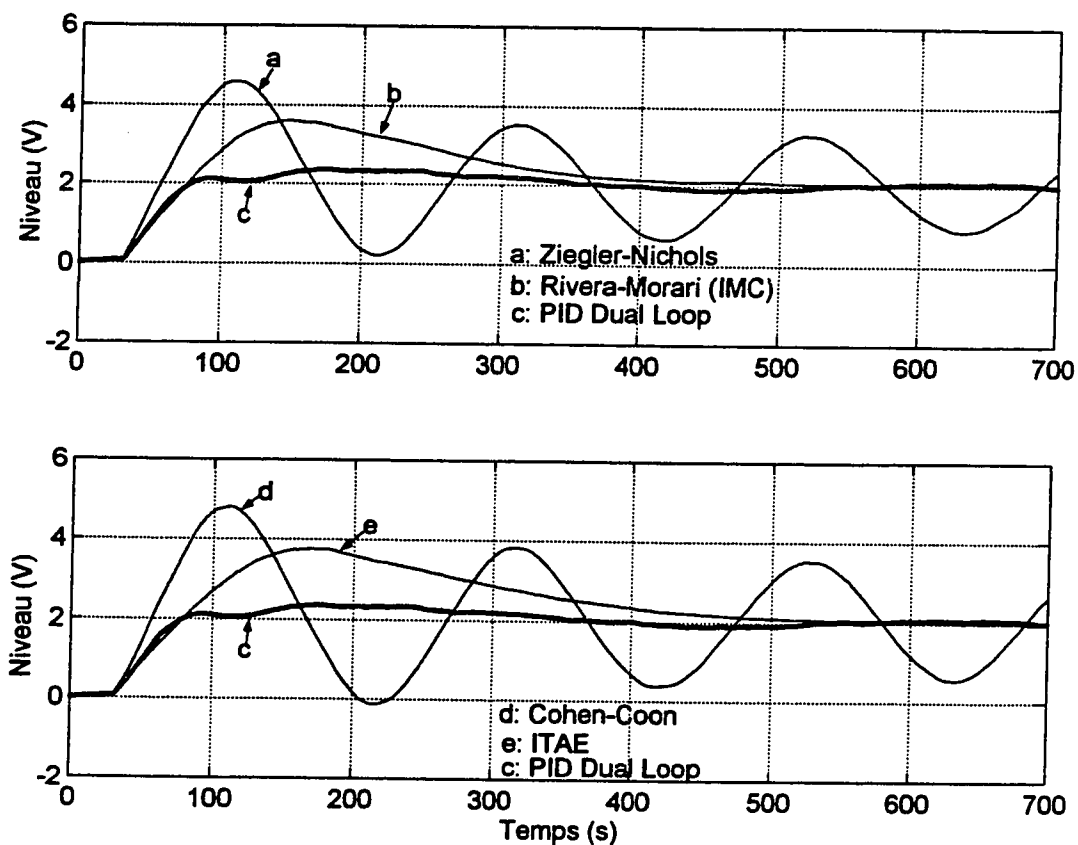


Figure 3.3: Régulation de niveau. Comparaison de différentes techniques de réglage

Résultats et commentaires: Les réglages de Ziegler-Nichols et Cohen-Coon donnent une réponse avec un dépassement voisin de 140% et des oscillations très marquées. Les réglages de Rivera-Morari et ITAE engendrent un dépassement voisin de 80%. Le temps de réponse à 5% est du même ordre de grandeur pour ces deux réglages (environ 400s pour le réglage IMC et environ 500s pour le réglage ITAE).

Avec le compensateur PID Dual Loop, il n'y a pas d'oscillations. Le dépassement est beaucoup plus faible (environ 15%) et le temps de réponse est de l'ordre de 350s.

3.4.2 Test de la robustesse

Objectif: Comparer le comportement du niveau du réservoir face à une perturbation avec les différents réglages des compensateurs PI et PI Dual Loop.

Modalités: Les modalités sont les mêmes qu'au paragraphe précédent sauf qu'on ajoute ici une perturbation sur la commande de la vanne d'eau froide (ajout d'un échelon d'un volt à $t=500s$).

Résultats et commentaires: La perturbation entretient les oscillations des réponses correspondant aux réglages de Cohen-Coon et de Ziegler et Nichols. Dans le cas des autres réglages, la perturbation est introduite après que le régime permanent soit atteint. Avec les réglages de Rivera-Morari et ITAE, le système retrouve son régime stationnaire en 500s environ et le dépassement engendré par la perturbation est de l'ordre de 90%.

Le PID Dual Loop limite ce dépassement à 40%, mais le temps de réponse augmente par rapport au réglage de Rivera-Morari (il subsiste une erreur de 7% 500s après l'introduction de la perturbation).

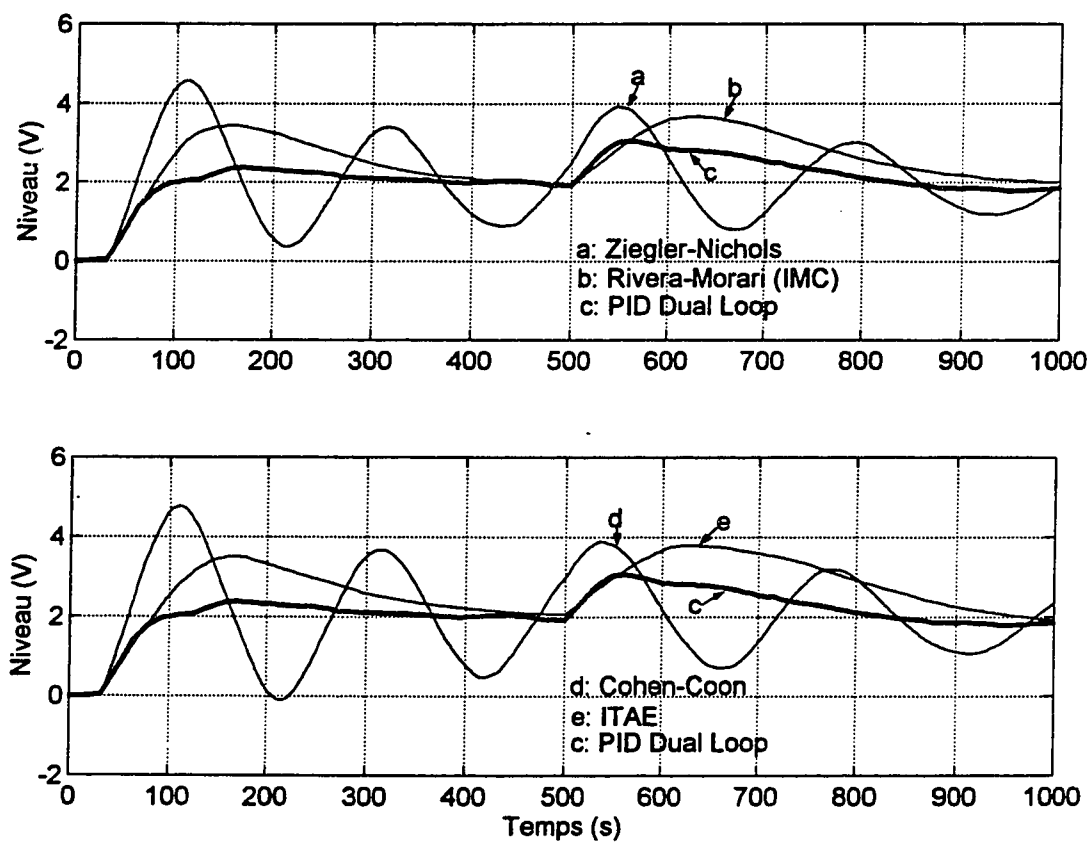


Figure 3.4: Test de la robustesse avec différents réglages.

3.5 Test sur le contrôle de la température seule

3.5.1 Asservissement de la température seule

Objectif: Mise en évidence des différences de performances de l'asservissement de température pour plusieurs types de réglage des compensateurs PI et PI Dual Loop décrits au paragraphe 3.2.

Modalités: Le principe est le même que pour l'asservissement de niveau sauf qu'ici on n'introduit pas de retard sur la commande car la température a un retard naturel de 20s. Lorsqu'on utilise les deux boucles d'asservissement simultanément (niveau et température, cf paragraphe 3.6), le retard sur la commande doit être le même pour les deux boucles. On se limite donc à un retard de 30s, le temps d'homogénéisation introduit déjà un retard naturel de 20s.

Pour ces tests, on ouvre le robinet de sortie au point fixé, puis on choisit le point de repos pour les débits d'eau froide et d'eau chaude de façon à stabiliser le niveau à 0.5m (0V) et la température à 20°C (mesure de 2V). On peut ensuite lancer le logiciel pint_eds.m pour le contrôle de température. La valeur de référence en température est fixée ici à 32°C (consigne de 5V). Les jeux de gains utilisés sont présentés au tableau 3.2.

Tableau 3.2: Paramètres des compensateurs PI et PID Dual Loop pour la régulation de température

Paramètres	Ziegler-Nichols	Cohen-Coon	Rivera-Morari (IMC), $\lambda = 80$	ITAE	PID Dual Loop
K_{21}	10.89	10.96	3.03	5.64	3.63
K_{22}	0	0	0	0	20.87
K_{23}	0.164	0.187	0.01	0.0189	0.02
K_{24}	0	0	0	0	0
A_{21}	0	0	0	0	61.74
A_{22}	0	0	0	0	2205

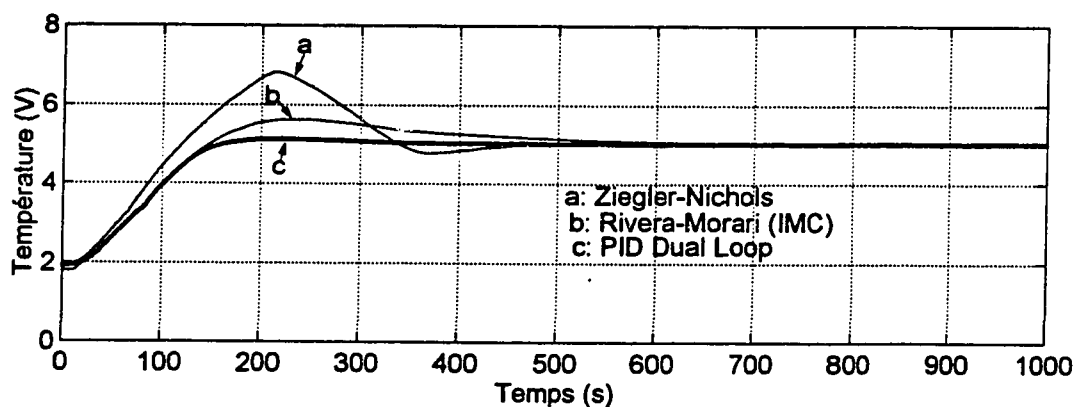


Figure 3.5a: Régulation de température. Comparaison de différentes techniques de réglage

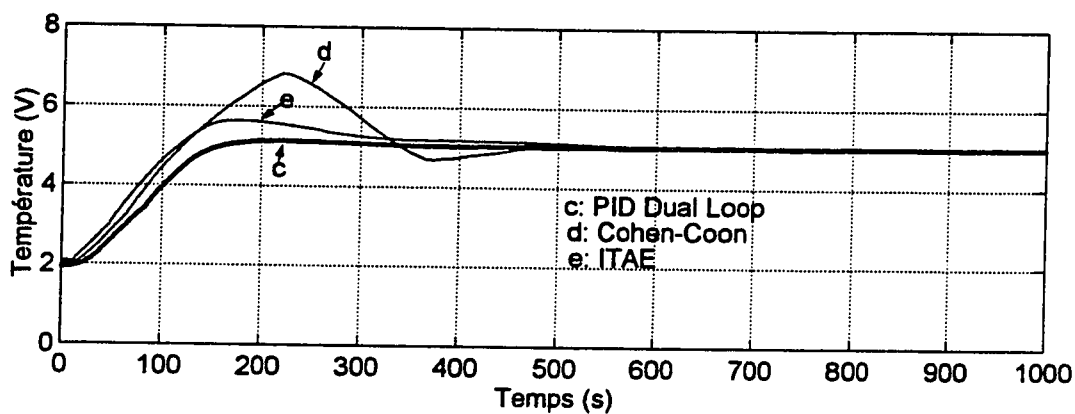


Figure 3.5b: Régulation de température. Comparaison de différentes techniques de réglage

Résultats et commentaires: La réponse en température avec les réglages de Ziegler-Nichols et de Cohen-Coon présente un premier dépassement de plus de 60% et une

légère oscillation. Le temps de réponse à 5% est pour ces réglages de l'ordre de 450s. Les résultats obtenus avec les 3 autres réglages se ressemblent. Néanmoins, le PID Dual Loop permet de réduire le premier dépassement (5% au lieu de 16% pour ITAE et IMC) et le temps de réponse à 5% (200s au lieu de 450s).

D'autre part, le niveau a beaucoup augmenté lors de ces expériences. Cela est dû à la saturation introduite par la vanne d'eau froide: lorsqu'on veut réchauffer le réservoir, le compensateur ouvre la vanne d'eau chaude et ferme la vanne d'eau froide en conséquence pour ne pas perturber le niveau. On obtient ainsi des tensions de commandes négatives pour la vanne d'eau froide mais ces commandes ne sont pas toujours réalisables. Par exemple, si la tension de commande de la vanne d'eau froide est initialement de 2V (valeur de repos, c'est-à-dire 0V en sortie de l'ordinateur), la vanne est fermée dès que la commande en sortie de l'ordinateur est inférieure à -2V; mais, en dessous de cette valeur, elle ne peut pas se fermer davantage et le niveau augmente car on ajoute beaucoup d'eau chaude en ne 'retirant' pas assez d'eau froide.

La figure ci-dessous présente les commandes envoyées par l'ordinateur aux vannes d'eau froide et d'eau chaude dans le cas des réglages de Rivera-Morari et de Cohen-Coon.

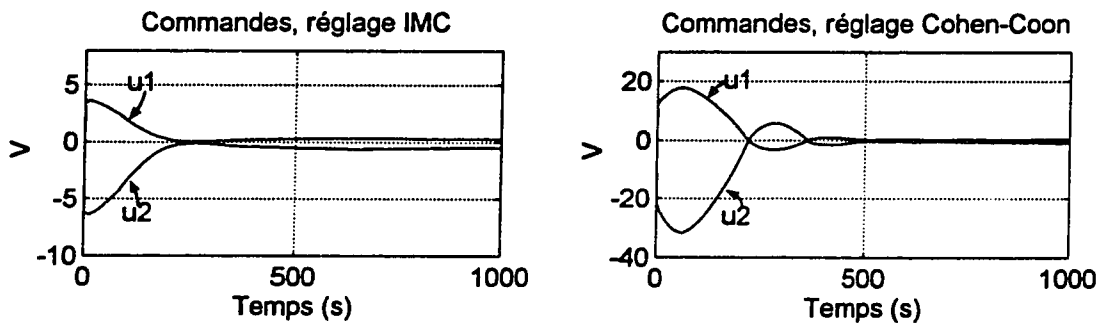


Figure 3.6: Commandes des vannes avec les réglages de Rivera-Morari et de Cohen-Coon

3.5.2 Test de la robustesse

Objectif: Comparer dans les conditions d'opérations nominales le comportement de la température du réservoir face à une perturbation avec un compensateur PID Dual Loop et un compensateur PI réglé avec la méthode IMC. Seuls ces deux compensateurs sont testés car ils se sont montrés les plus performants lors du test précédent.

Modalités: Ce sont les mêmes qu'au paragraphe précédent sauf qu'on introduit une perturbation à $t=750s$. Cette perturbation de $-0.54V$ sur la vanne d'eau chaude et de $0.96V$ sur la vanne d'eau froide est celle qui a été décrite au paragraphe 3.3.1.

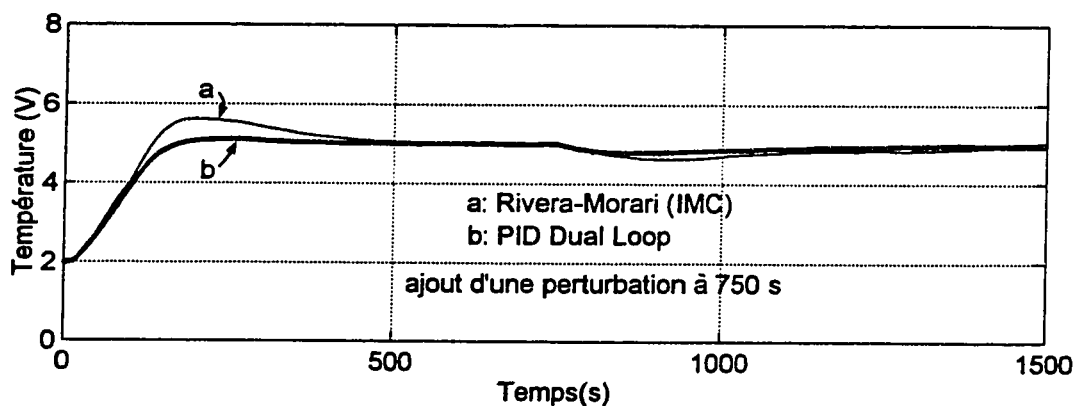


Figure 3.7: Comparaison de la robustesse avec deux compensateurs différents

Résultats et commentaires: Le PID Dual Loop réagit un peu mieux que le PID IMC à cette perturbation car il diminue le dépassement qu'elle provoque. Le régime permanent est rejoint en 500s environ. Ce test de robustesse est assez significatif puisque la perturbation introduite est importante (en régime permanent, les valeurs des commandes sont respectivement de 0.6V pour la vanne d'eau chaude et 1.7V pour la vanne d'eau froide).

3.6 Test des deux boucles scalaires

3.6.1 Asservissement simultané du niveau et de la température

Objectif: Illustrer le comportement des deux compensateurs utilisés simultanément et avec les réglages obtenus lors de l'étude séparée de chacune des boucles.

Modalités des tests: Les expériences sont réalisées sous les mêmes conditions qu'aux parties 3.4 et 3.5 sauf que maintenant les deux boucles fonctionnent de façon simultanée. On conserve les valeurs de gains obtenues pour les régulations séparées. On utilise le même type de réglage pour chacune des boucles. Par exemple, Ziegler et Nichols pour le niveau et Ziegler et Nichols pour la température ou IMC pour le niveau et IMC pour la température.

On utilise le découpleur de la figure 3.2. Comme dans toutes les expériences précédentes $\beta_1 = 0.36$ et $\beta_2 = 0.64$, en plus ici $\alpha_r = 1.7$.

On introduit un retard de 30s sur les commandes. Ce retard agit sur la boucle de niveau et sur la boucle de température. La boucle de température a un retard naturel de 20s plus le retard de 30s introduit artificiellement.

Le robinet de sortie est ouvert à son point fixé et on fixe le débit de repos pour les vannes d'eau chaude et d'eau froide de manière à stabiliser le niveau à 0.5m (0V) et obtenir la température de repos désirée 20°C (2V).

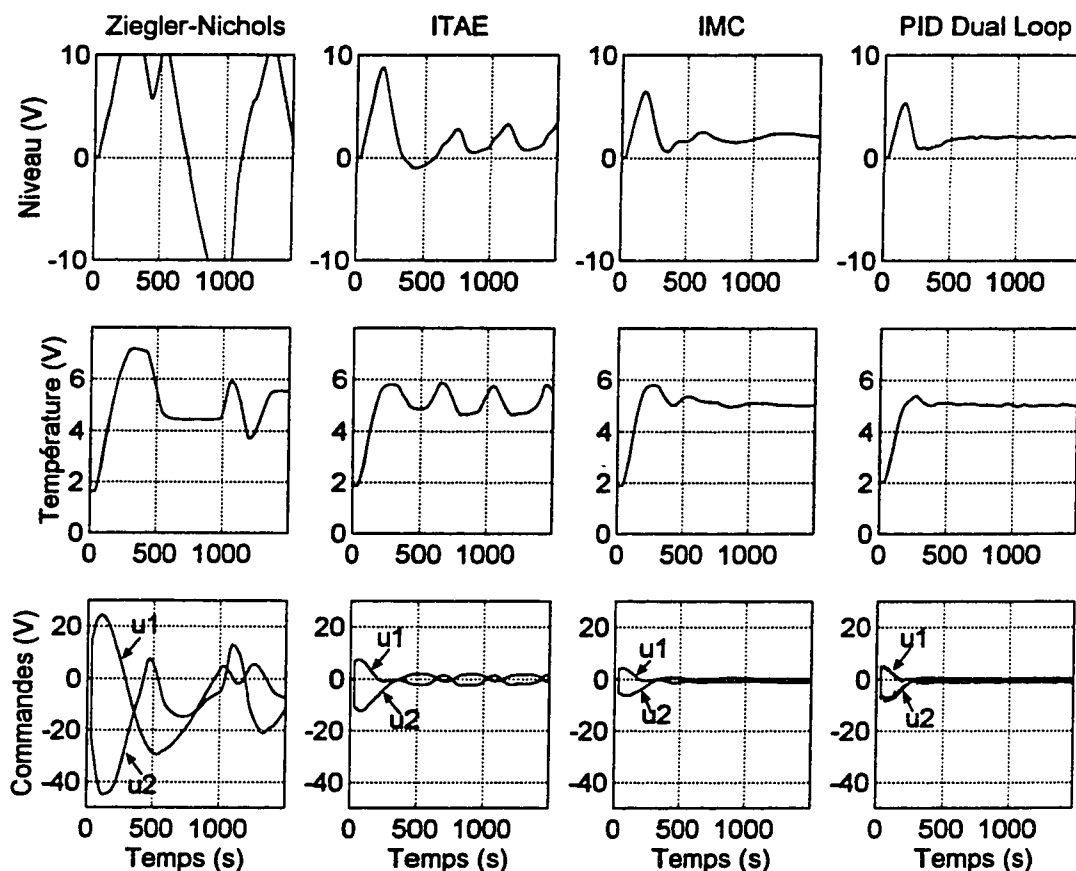


Figure 3.8: Fonctionnement simultané des deux asservissements

Résultats et commentaires: Les réglages de Ziegler-Nichols et ITAE provoquent des oscillations qui ne permettent pas d'atteindre la consigne de niveau.

Le PID Dual Loop et le réglage de Rivera-Morari (IMC) pour le PI permettent d'atteindre les consignes de niveau et de température en un temps raisonnable. On s'aperçoit que les performances en suivi de consignes sont très proches pour ces deux compensateurs lorsqu'on utilise les deux boucles simultanément.

3.6.2 Test de la robustesse

Objectif: Illustrer le comportement du niveau et de la température face à une perturbation en utilisant les deux boucles scalaires avec deux réglages différents: PI de Rivera-Morari et PID Dual Loop.

Modalités: Ce sont les mêmes qu'au paragraphe précédent sauf qu'on introduit la perturbation de 1V décrite au paragraphe 3.3.1.

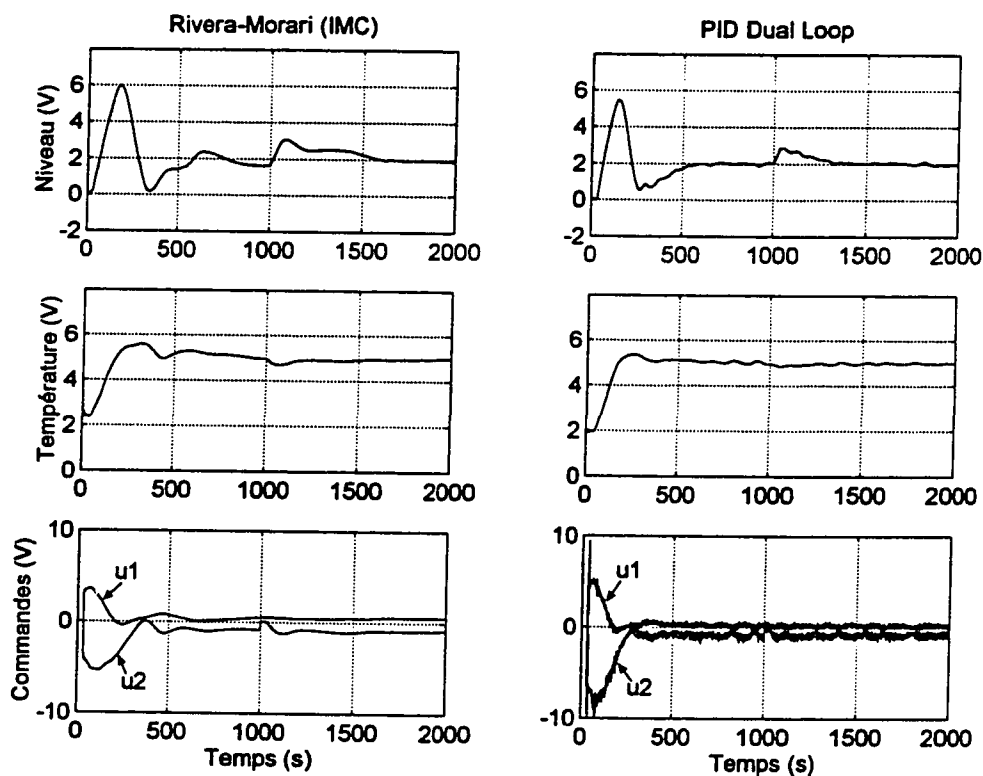


Figure 3.9: Niveau et température face à une perturbation

Résultats et commentaires: Le réglage du PID Dual Loop est meilleur que le réglage IMC pour réguler cette perturbation. En effet la perturbation n'engendre qu'un dépassement de 50% pour le réglage du PID Dual Loop alors que dans le cas IMC on obtient un dépassement de 70%. De plus, le PID Dual Loop permet de rejoindre beaucoup plus rapidement le régime stationnaire (en 400s au lieu de plus de 1000s).

3.7 Difficultés techniques

Les problèmes rencontrés lors de l'utilisation simultanée des deux boucles sont essentiellement liés aux limitations sur les vannes (saturation en fermeture et parfois en ouverture), à la pression d'air et à la variation importante des températures d'eau chaude et d'eau froide. Ces éléments font qu'il n'est pas toujours facile de mener à bien des expériences dont les résultats puissent être comparés d'une façon cohérente. Pour surmonter ce problème on a fait en sorte que la variable α_r devienne une fonction de la température.

Par ailleurs, il faut noter que si l'on disposait d'un débit de sortie plus élevé, les valeurs de repos des commandes de vannes seraient plus élevées et on aurait peut-être plus de marge de manœuvre pour diminuer la commande d'eau froide dans le cas de l'asservissement de la température. On aurait ainsi moins de problèmes avec la saturation en fermeture de la vanne d'eau froide.

3.8 Conclusion sur les expériences

L'objectif de ce chapitre, le réglage des compensateurs PID classique et PID Dual Loop pour l'asservissement du niveau et de la température d'un réservoir d'eau, a été atteint. Le banc d'essai a permis d'effectuer des tests expérimentaux qui illustrent bien les différentes techniques de réglage et leurs performances.

Comme prévu, le retard de 30s introduit sur la commande rend plus difficile l'asservissement du niveau que l'asservissement de la température. Ainsi, dans le cas des boucles scalaires, toutes les techniques de réglage donnent un temps de réponse acceptable sauf Cohen-Coon et Ziegler et Nichols pour le niveau. Les techniques ITAE et IMC permettent d'obtenir des compensateurs assez robustes car, outre la perturbation de 1V qui est ajoutée volontairement, le système est soumis à de nombreuses perturbations telles que des variations de température d'eau chaude et d'eau froide, des variations de pression dans le réseau de distribution d'eau et des difficultés d'ajustement du régime de repos. Lorsqu'on utilise un seul compensateur, la structure PID Dual Loop permet d'améliorer rapidement les performances (temps de réponse, dépassement, robustesse) obtenues avec le meilleur des réglages du PID classique. Lorsqu'on utilise les deux boucles simultanément, le PID Dual Loop offre des performances assez voisines de celles du PID classique réglé avec la méthode IMC.

La procédure de réglage des gains du compensateur PID Dual Loop est claire et facile à implanter. Cette procédure prend une dizaine de minutes en fonction de l'expérience et de la réponse entrée-sortie. Par rapport aux autres techniques classiques mentionnées précédemment, le réglage des gains du compensateur PID Dual est plus structuré et le temps de réglage est satisfaisant.

Dans le prochain chapitre, on va développer un module d'adaptation automatique pour régler les gains du compensateur PID Dual Loop. Le module sera adapté au banc d'essai pour l'asservissement du niveau et de la température.

CHAPITRE IV

ÉTUDE EXPÉRIMENTALE DU COMPENSATEUR PID DUAL LOOP

AVEC UN MODULE D'ADAPTATION AUTOMATIQUE

4.1 Introduction

L'objectif de ce chapitre est double: il s'agit d'abord de mettre au point un module permettant le réglage automatique (auto-tuning) des gains des compensateurs PID Dual Loop contrôlant le niveau et la température dans le réservoir d'eau présenté au chapitre II (Cornieles et Bougeret, 1997). Ce module permettra de régler les gains de l'un ou l'autre des compensateurs (contrôle du niveau ou de la température) ou bien des deux compensateurs (contrôle du niveau et de la température) en utilisant la méthode et la procédure développées au chapitre I à l'aide de l'organigramme présenté à l'annexe 5 et du programme de l'annexe 10. Deuxièmement, les résultats expérimentaux obtenus permettront ensuite de dire dans quelle mesure les propriétés du PID Dual Loop (possibilité de régler indépendamment la dynamique et la robustesse) restent valables dans le cas d'un système de dynamique inconnue présentant un retard, des saturations et des perturbations extérieures du banc d'essai (variation de la température d'eau d'alimentation du réservoir, de la pression d'air et de perturbations électriques).

4.2 Rappel des structures du banc d'essai et du compensateur PID Dual Loop

4.2.1 Structure générale de l'asservissement

Les commandes sont les tensions U_1 et U_2 qui permettent par l'intermédiaire d'un découpleur de varier les ouvertures des vannes pneumatiques d'eau chaude et d'eau froide arrivant dans le réservoir. Les mesures sont les tensions pv_1 et pv_2 fournies par les capteurs et correspondent au niveau et à la température de l'eau sortant du réservoir. Normalement (avec le module de calcul automatique des gains), le banc d'essai a la structure décrite à la figure 4.1:

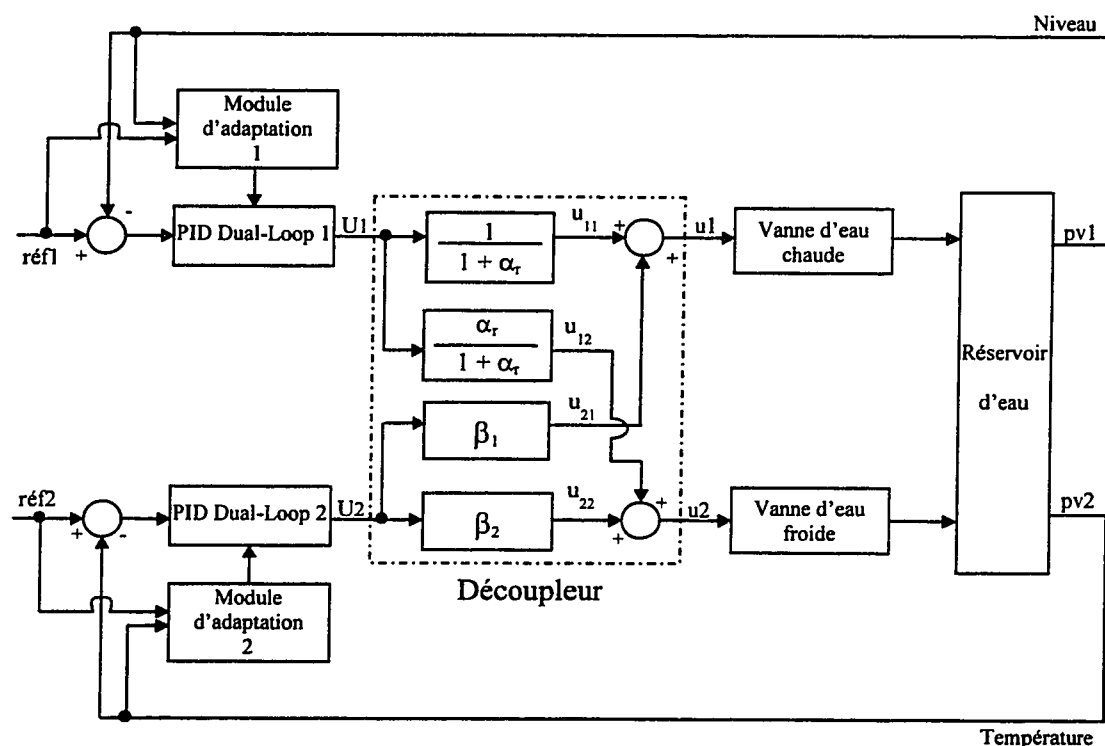


Figure 4.1: Structure générale de l'asservissement

4.3 Résultats expérimentaux

On présente maintenant les tests réalisés sur le réservoir avec le programme de recherche automatique de gains pintpri2.m et raz.m (avec compensateur proportionnel) pour la remise à zéro.

Seront également présentés les tests de robustesse effectués avec différentes valeurs de K_3 . Ces tests (réalisés avec le logiciel pint_eds.m développé antérieurement (Cornieles et Bougeret, 1997)) permettront de vérifier que l'augmentation de K_3 permet d'augmenter la robustesse sans détériorer la réponse. Les tests ont été faits avec un retard de 30s, sauf le test de la robustesse pour la température qui a été fait avec un retard de 20s.

4.3.1 Niveau

Objectif: Il s'agit de vérifier que le programme de réglage automatique est capable de donner un jeu de gains convenable pour l'asservissement du niveau uniquement.

Modalités: On ajuste les débits initiaux d'eau chaude et d'eau froide pour obtenir l'équilibre du système (niveau constant). Dans le logiciel pintpri2.m, on choisit $K_{11}=0.7$ (valeur initiale du gain proportionnel), un retard de 30s, un temps de premier dépassement désiré de 140s, une amplitude relative de premier dépassement de 10%

pour la première étape du réglage et une amplitude relative de premier dépassement de 30% pour la seconde étape du réglage. On présente ici deux expériences réalisées avec ces conditions mais avec des incréments incK13 différents: 0.003 pour la première et 0.001 pour la seconde. Chaque test dure 750s et chaque remise à zéro dure 1000s.

Résultats:

• Première expérience: $\text{incK13}=0.003$

On présente ici les résultats successifs des 7 tests effectués automatiquement par le programme `pintpri2.m`.

Tableau 4.1: Résultats de la première expérience concernant le niveau

n°	K11	K12	K13	A11	A12	tpdn (s)	depn	oscil
1	0.70	0	0	0	0	128	0.28	-
2	0.70	3.90	0	0	0	136	0.22	-
3	0.70	8.56	0	0	0	165	0.07	-
4	0.85	12.2	0	0	0	74.5	0.03	-
5	0.85	12.2	0.003	37.3	347	101	0.27	-
6	0.85	12.2	0.006	37.3	347	94	0.29	-
7	0.85	12.2	0.009	37.3	347	91	0.47	-

On remarque que la première étape du réglage correspond aux 4 premiers tests car les critères de premier dépassement sont respectés au quatrième test ($74.5s < 140s$ et $3\% < 10\%$). La seconde étape s'arrête au septième test car l'amplitude du premier dépassement y est supérieure à l'amplitude désirée ($47\% > 30\%$). On remarque que l'erreur en régime permanent diminue avec l'incrément de K_{13} . Les gains définitifs trouvés correspondent donc au sixième test:

$$K_{11}=0.85 \quad K_{12}=12.2 \quad K_{13}=0.006 \quad A_{11}=37.3 \quad A_{12}=347$$

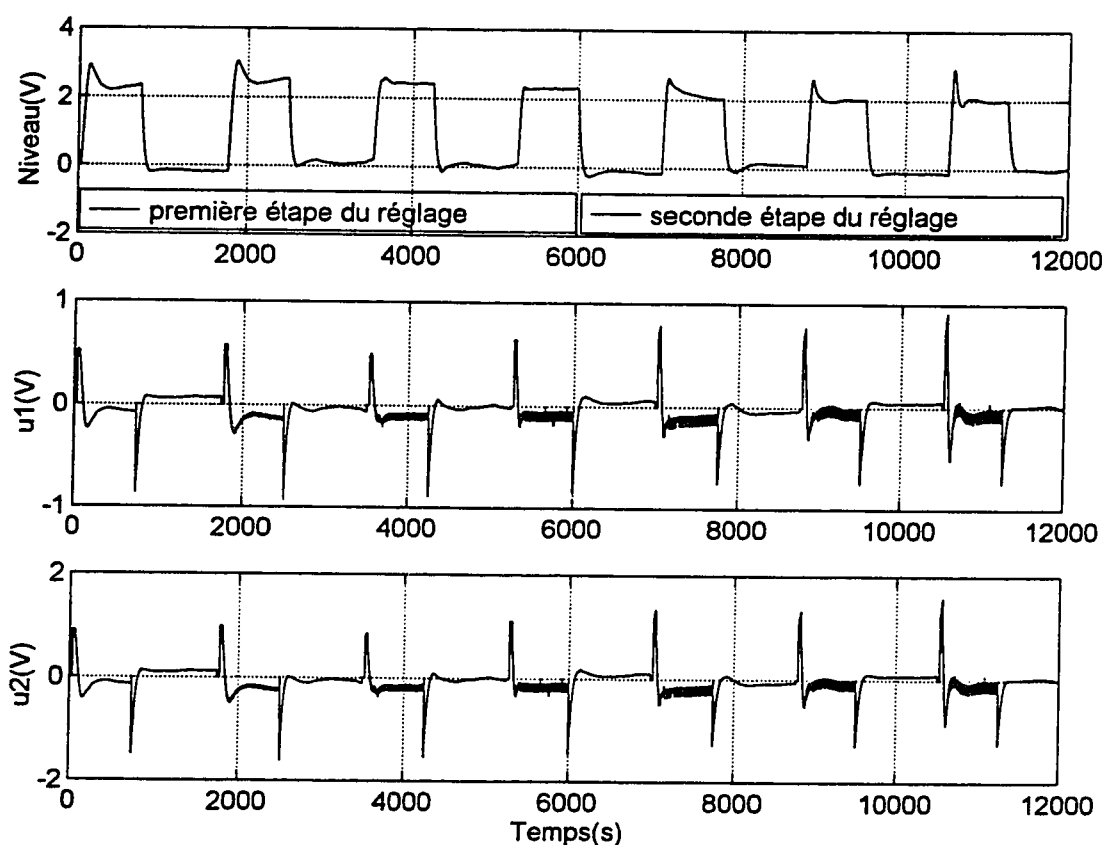


Figure 4.2: Résultat global du réglage automatique (niveau et commandes) des 7 tests

Sur la figure 4.3 sont superposées les réponses du système aux 4 tests de la première étape de réglage (réglage de K_{11} et de K_{12}).

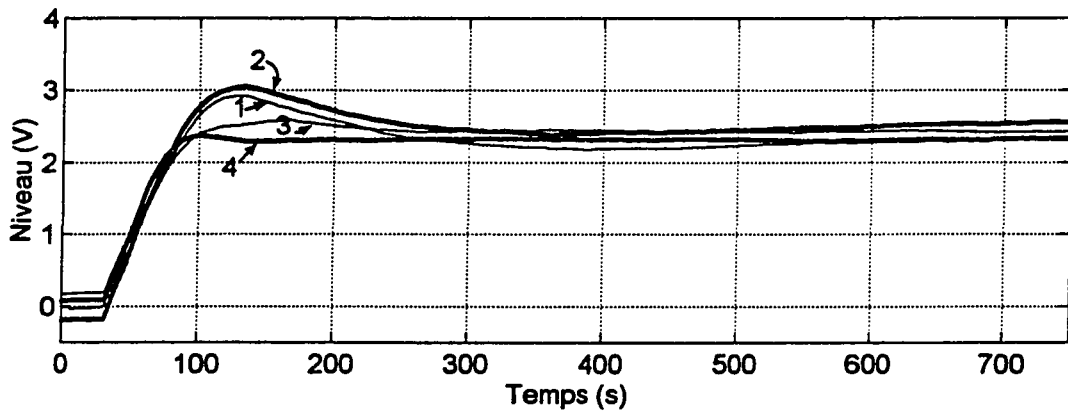


Figure 4.3: Évolution de la réponse lors de la première étape du réglage (niveau)

• Test de la robustesse

Pour tester la robustesse du système, on ajoute un échelon de perturbation de 1V sur la commande de la vanne d'eau froide à $t=500s$ (quand le régime stationnaire est établi). On présente ici le comportement du niveau du réservoir lorsqu'il est asservi par deux jeux de gains différents. Le premier jeu de gains utilisé est celui trouvé par l'auto-tuning. Le second jeu est le même que le premier sauf que le gain K_{13} a été diminué jusqu'à $incK_{13}$ ($K_{13}=0.003$).

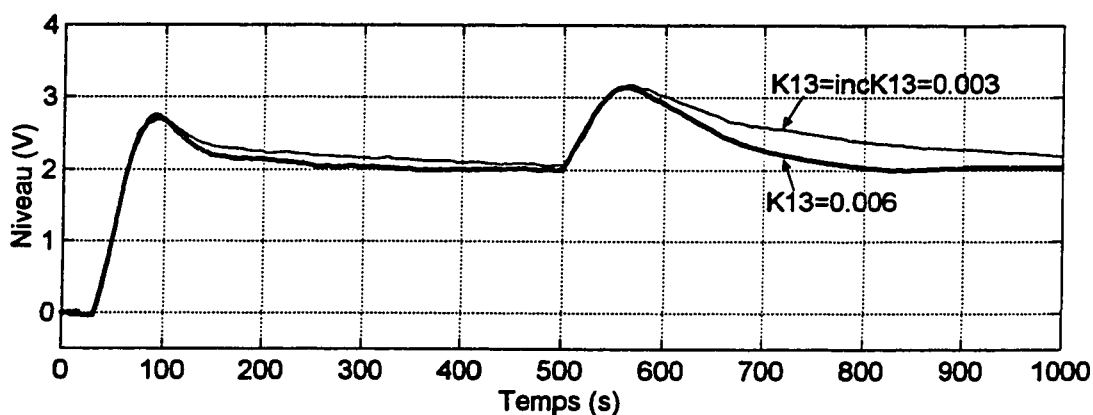


Figure 4.4: Évolution de la robustesse avec l'augmentation de K13 (niveau)

• Analyse des résultats

À la figure 4.2, on constate que les gains K11 et K12 sont modifiés progressivement lors de la première étape de réglage de façon à satisfaire les critères fixés par l'utilisateur. Bien sûr, ces critères sont raisonnables et s'inspirent des résultats obtenus avec un réglage manuel (Cornieles et Bougeret, 1997). En effet, nous avons pu vérifier expérimentalement que des critères trop sévères sur le temps du premier dépassement mènent à des jeux de gains qui provoquent des oscillations dans la réponse.

En ce qui concerne le réglage de K13, on constate au tableau 4.1 que plus K13 est élevé, plus l'amplitude de premier dépassement est grande. Cette constatation montre que les

approximations (1.22) et (1.23) ont des limites. Malgré tout, le temps du premier dépassement est assez bien conservé et ces approximations permettent d'obtenir une méthode de réglage qui donne un jeu de gains satisfaisant.

D'autre part, la figure 4.4 montre que plus K_{13} est élevé, plus le système est robuste, ce qui va bien dans le sens de l'équation (1.21).

Puisque la seconde étape de réglage ne comprend ici que 3 tests, une deuxième expérience sur le niveau va être menée avec les mêmes conditions logicielles initiales mais avec un incrément $\text{inc}K_{13}$ plus petit. Cela permettra peut-être d'affiner le réglage de K_{13} .

•Deuxième expérience: $\text{inc}K_{13}=0.001$

On présente ici les résultats successifs des 15 tests effectués automatiquement par le programme `pintpri2.m`.

Tableau 4.2: Résultats de la deuxième expérience concernant le niveau

n°	K11	K12	K13	A11	A12	Tpdn (s)	depn	oscil
1	0.7	0	0	0	0	155	0.35	-
2	0.85	6.62	0	0	0	113	0.31	-
3	0.85	10.6	0	0	0	104	0.13	-
4	0.85	15.7	0	0	0	74	0.01	-
5	0.85	15.7	0.001	38.5	371	71	0.03	-
6	0.85	15.7	0.002	38.5	371	94	0.08	-
7	0.85	15.7	0.003	38.5	371	86	0.11	-
8	0.85	15.7	0.004	38.5	371	91	0.16	-
9	0.85	15.7	0.005	38.5	371	81	0.16	-
10	0.85	15.7	0.006	38.5	371	80	0.23	-
11	0.85	15.7	0.007	38.5	371	80	0.21	-
12	0.85	15.7	0.008	38.5	371	77	0.25	-
13	0.85	15.7	0.009	38.5	371	78	0.25	-
14	0.85	15.7	0.010	38.5	371	75	0.29	-
15	0.85	15.7	0.011	38.5	371	74	0.35	-

On remarque que la première étape du réglage correspond aux 4 premiers tests car les critères du premier dépassement sont respectés au quatrième test ($74s < 140s$ et $1\% < 10\%$).

La seconde étape s'arrête au 15^{ème} test car l'amplitude du premier dépassement y est supérieure à l'amplitude désirée ($35\% > 30\%$). Les gains définitifs trouvés correspondent donc au quatorzième test. Ce sont:

$K11=0.85$ $K12=15.7$ $K13=0.01$ $A11=38.5$ $A12=371$

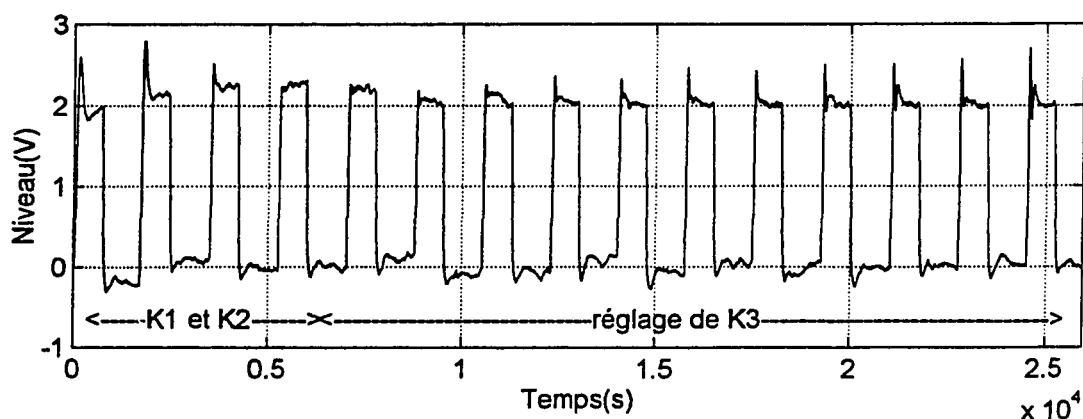


Figure 4.5: Résultat global du réglage automatique (niveau) des 15 tests

À la figure 4.6 sont superposées les réponses du système aux 4 tests de l'étape de réglage de $K11$ et $K12$.

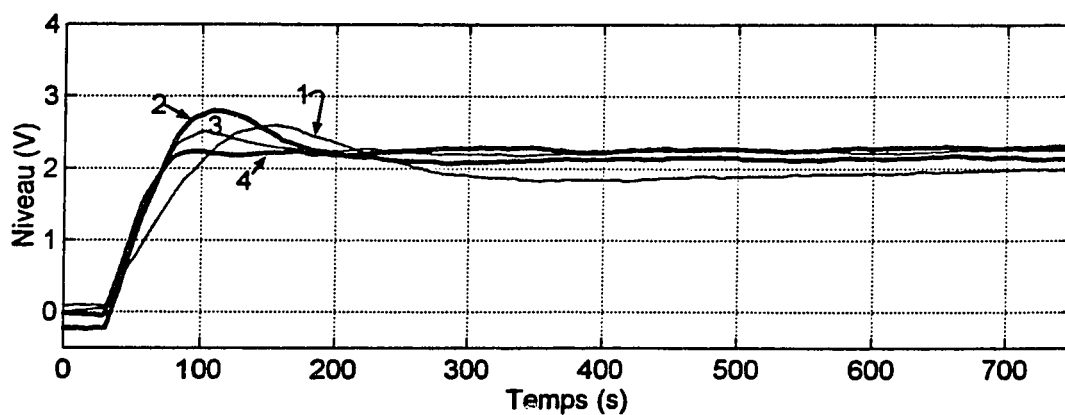


Figure 4.6: Évolution de la réponse lors de la première étape du réglage (niveau)

Remarque: Les figures 4.5 et 4.6 font voir que le programme raz.m remet le niveau à zéro après chaque test. Il le fait au moyen d'un compensateur proportionnel qui a un gain quelconque.

• Test de la robustesse

La perturbation utilisée pour ce test est la même que celle utilisée pour la première expérience. Le premier jeu de gains utilisé est celui trouvé par l'auto-tuning. Le second jeu est le même que le premier, sauf que le gain K_{13} a été diminué jusqu'à $incK_{13}$ ($K_{13}=0.001$). L'erreur en régime permanent diminue avec l'incrément de K_{13} .

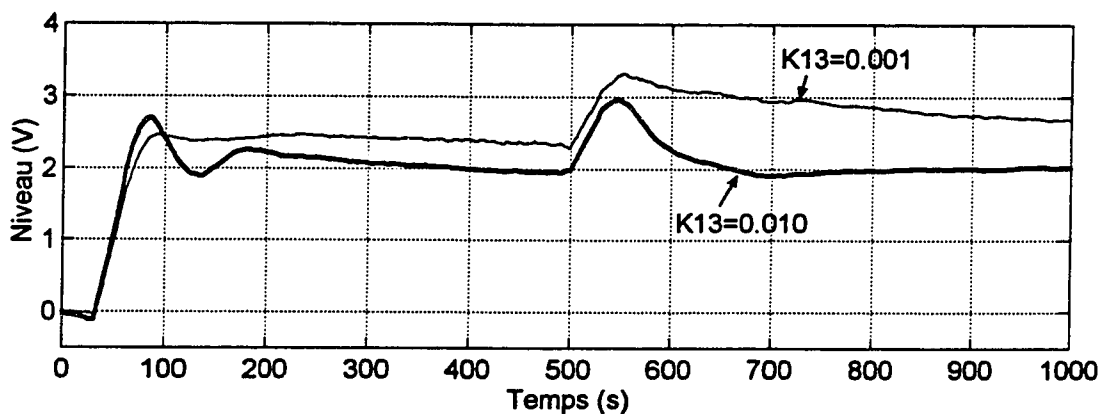


Figure 4.7: Évolution de la robustesse avec l'augmentation de K_{13} (niveau)

• Analyse des résultats

L'analyse est globalement la même que pour la première expérience sur le niveau. On se rend compte qu'avec les mêmes conditions logicielles initiales, on obtient un réglage différent, notamment un gain K_{13} plus élevé. Cela est, bien sûr, dû à une température et à une pression légèrement différentes pour chaque alimentation en eau (expérience faite pendant la nuit). L'influence de K_{13} sur la robustesse du système est encore plus flagrante à la figure 4.7 où le rapport entre les gains K_{13} testés est encore plus grand qu'à la figure 4.4 (rapport 10 contre rapport 2).

4.3.2 Température

Objectif: Il s'agit de vérifier que le programme de réglage automatique est capable de donner un jeu de gains convenable pour l'asservissement de température.

Modalités: On ajuste les débits initiaux d'eau chaude et d'eau froide pour obtenir l'équilibre du système (niveau constant à 0V et température à 2V). Dans l'interface utilisateur, on choisit $K_{21}=2$ (valeur initiale du gain proportionnel), un temps de premier dépassement désiré de 200s, une amplitude relative de premier dépassement de 10% pour la première étape du réglage, une amplitude relative de premier dépassement de 20% pour la seconde étape du réglage et un incrément $\text{inc}K_{23}$ égal à 0.01. Chaque test dure 1500s et chaque remise à zéro dure 1250s.

Résultats:

On présente ici les résultats successifs des 8 tests effectués automatiquement par le programme pintpri2.m.

Tableau 4.3: Résultats de l'expérience concernant la température

n°	K21	K22	K23	A21	A22	Tpdt (s)	dept	oscil
1	2	0	0	55	3072	201	0.17	-
2	2.42	35	0	75	1416	151	0.05	-
3	2.42	35	0.01	75	1416	187	0.01	-
4	2.42	35	0.02	75	1416	351	0.06	-
5	2.42	35	0.03	75	1416	287	0.11	-
6	2.42	35	0.04	75	1416	168	0.13	-
7	2.42	35	0.05	75	1416	203	0.11	-
8	2.42	35	0.06	75	1416	200	0.13	oui

On remarque que la première étape du réglage correspond aux 2 premiers tests car les critères de premier dépassement sont respectés au deuxième test ($151s < 200s$ et $5\% < 10\%$).

La seconde étape s'arrête au huitième test car le programme détecte la présence des oscillations.

Les gains définitifs trouvés correspondent donc au septième test. Ce sont:

$$K_{21}=2.42 \quad K_{22}=35 \quad K_{23}=0.05 \quad A_{21}=75 \quad A_{22}=1416$$

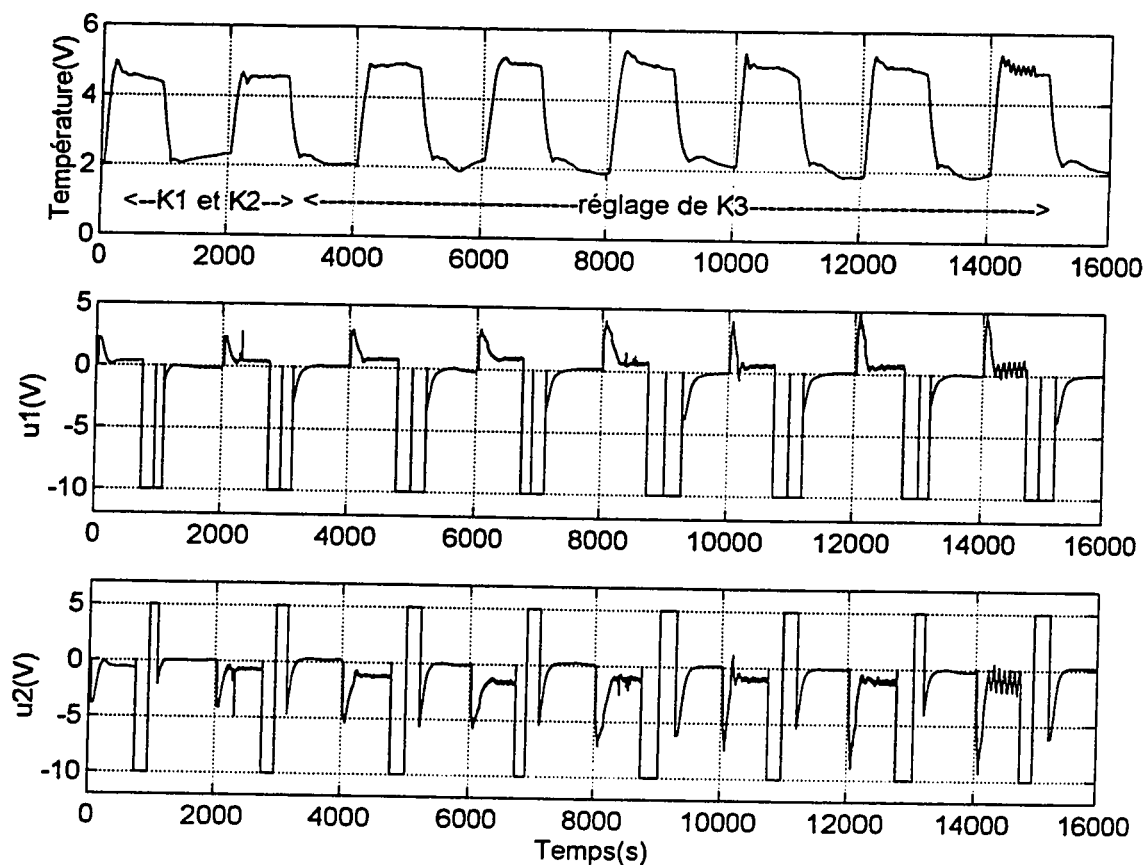


Figure 4.8: Résultat global du réglage automatique (température et commandes)

Sur la figure 4.9 sont superposées les réponses du système aux 2 tests de l'étape de réglage de K_{21} et K_{22} .

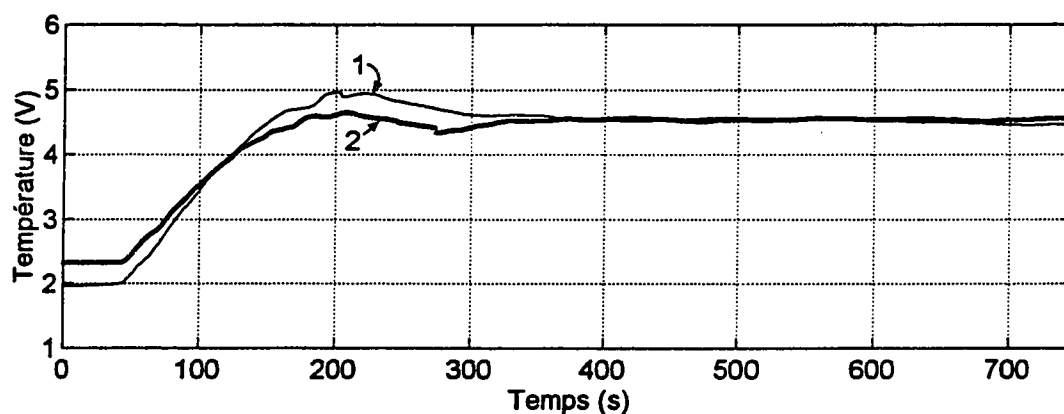


Figure 4.9: Évolution de la réponse lors de la première étape du réglage (température)

• Test de la robustesse

On présente le comportement de la température face à un échelon de perturbation de 1.92V sur la commande de la vanne d'eau froide et de -1.08V sur la commande de la vanne d'eau chaude. Les valeurs numériques de cette perturbation ont été calculées en fonction des gains des vannes pneumatiques afin de ne pas trop perturber le niveau de l'eau dans le réservoir.

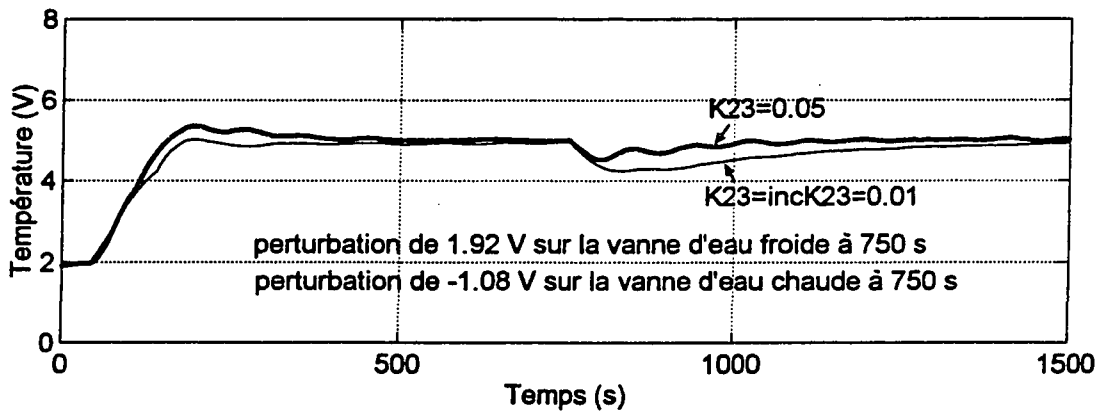


Figure 4.10: Évolution de la robustesse avec l'augmentation de K_{23} (température)

• Analyse des résultats

La première étape de réglage n'est pas très longue (seulement 2 tests). Peut-être aurait-on dû être un peu plus sévère sur les critères du premier dépassement mais les critères choisis semblaient déjà meilleurs que les résultats obtenus avec un réglage manuel du Dual Loop (Cornieles et Bougeret, 1997). En tout cas, on vérifie que le programme analyse bien la réponse du système au premier test (système asservi par un simple compensateur proportionnel).

À propos du gain proportionnel du premier test, signalons qu'il ne faut pas initialiser le gain proportionnel K_{21} avec une valeur trop grande car sinon le programme calcule des

gains K_{21} et K_{22} qui provoquent des oscillations dans la réponse dès les premières itérations. Cela paraît normal puisque le programme est incapable de diminuer la valeur de K_{21} lors d'une itération (cf équation (1.30)).

En ce qui concerne le réglage de K_{23} , on constate au tableau 4.3 que l'introduction de K_{23} a plutôt tendance à faire augmenter le temps et l'amplitude du premier dépassement. Mais surtout, on s'aperçoit qu'une valeur trop grande pour K_{23} provoque des oscillations dans la réponse. Cette constatation rejoint celle qui a été faite pour les expériences sur le niveau, à savoir que les approximations (1.22) et (1.23) ne sont légitimes que pour des valeurs suffisamment faibles.

Malgré tout, ces approximations permettent d'obtenir une méthode de réglage qui donne un jeu de gains satisfaisant.

D'autre part, la figure 4.10 montre que, plus K_{23} est élevé, plus le système rejoint rapidement le régime stationnaire après l'introduction des échelons de perturbation. Cette constatation va bien dans le sens de l'équation (1.21).

4.3.3 Niveau et température

Objectif: Il s'agit de vérifier que le programme de réglage automatique est capable de donner un jeu de gains convenable pour l'asservissement du niveau et de la température.

Modalités: On ajuste les débits initiaux d'eau chaude et d'eau froide pour obtenir l'équilibre du système (niveau constant à 0V et température à 2V). Dans l'interface utilisateur, on choisit $K_{11}=0.7$ et $K_{21}=1.5$ (valeurs initiales des gains proportionnels pour le niveau et la température), un temps du premier dépassement désiré pour le niveau de 180s, une amplitude relative du premier dépassement de 20% pour la première étape du réglage, une amplitude relative du premier dépassement de 40% pour la seconde étape du réglage et un incrément $incK_{13}$ de 0.001. Pour la température, un temps de premier dépassement désiré de 180s, une amplitude relative du premier dépassement de 10% pour la première étape du réglage, une amplitude relative du premier dépassement de 20% pour la seconde étape du réglage et un incrément $incK_{23}$ de 0.003. La durée de chaque test est de 1000s et la durée de remise à zéro de 1250s.

Résultats:

On présente ici les résultats successifs des 10 tests effectués automatiquement par le programme `pintpri2.m`.

Tableau 4.4: Résultats de l'expérience concernant le niveau et la température

n°	K11	K12	K13	A11	A12	K21	K22	K23	A21	A22	Tpdn (s)	depn	tpdt (s)	dept	oscl
1	0.7	0	0	0	0	1.5	0	0	0	0	163	0.30	216	0.21	-
2	0.7	4.84	0	0	0	1.82	26.1	0	0	0	186	0.33	179	0.15	-
3	0.85	13.1	0	0	0	1.82	44.5	0	0	0	203	0.15	141	0.04	-
4	1.02	18.0	0	0	0	1.82	44.5	0	0	0	162	0.06	145	0.01	-
5	1.02	18.0	0.001	50.8	1455	1.82	44.5	0.003	72.3	1305	216	0.22	172	0.01	-
6	1.02	18.0	0.002	50.8	1455	1.82	44.5	0.006	72.3	1305	188	0.20	187	0.01	-
7	1.02	18.0	0.003	50.8	1455	1.82	44.5	0.009	72.3	1305	277	0.14	260	0.08	-
8	1.02	18.0	0.004	50.8	1455	1.82	44.5	0.012	72.3	1305	-	-	204	0.04	niveau
9	1.02	18.0	0.004	50.8	1455	1.82	44.5	0.009	72.3	1305	-	-	242	0.04	niveau
10	1.02	18.0	0.003	50.8	1455	1.82	44.5	0.012	72.3	1305	-	-	414	0.12	niveau

On remarque que la première étape du réglage correspond aux 4 premiers tests car les critères de premier dépassement pour le niveau sont respectés au quatrième test ($162s < 180s$ et $6\% < 20\%$) et aussi pour la température ($145s < 180s$ et $1\% < 10\%$). Du test 5 jusqu'au test 8, le programme augmente les valeurs de K13 et de K23. Suite aux oscillations rencontrées pendant le test 8, le programme essaie d'augmenter K13 sans augmenter K23 puis inversement (essais 9 et 10) mais la réponse en niveau présente des oscillations lors de ces deux essais. Les gains définitifs trouvés correspondent donc au septième test. Ce sont:

K11=1.02 K12=18 K13=0.003 A11=50.8 A12=1455 (pour le niveau)

K21=1.82 K22=44.5 K23=0.009 A21=72.3 A22=1305 (pour la température)

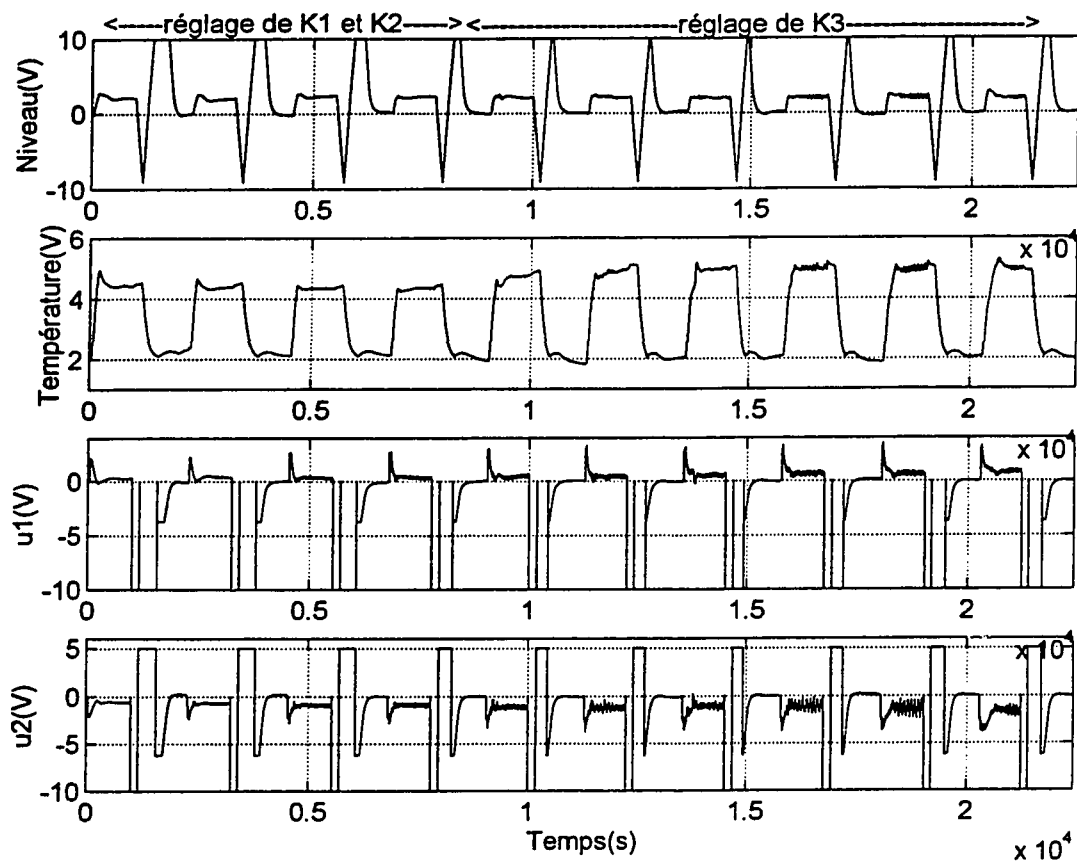


Figure 4.11: Résultat global du réglage automatique (commandes, niveau et température)

À la figure 4.12 et à la figure 4.13 sont superposées les réponses en niveau et en température lors des 4 tests de l'étape de réglage de K11, K12, K21 et K22.

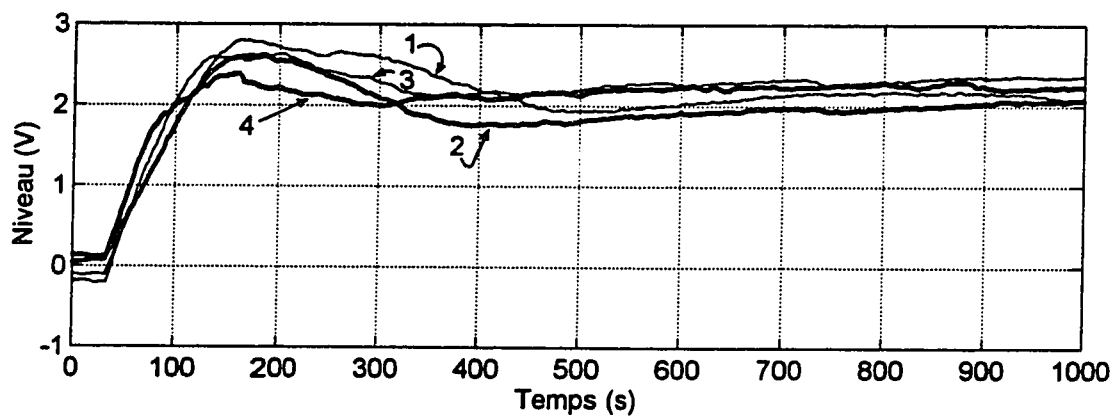


Figure 4.12: Évolution de la réponse en niveau lors de la première étape du réglage

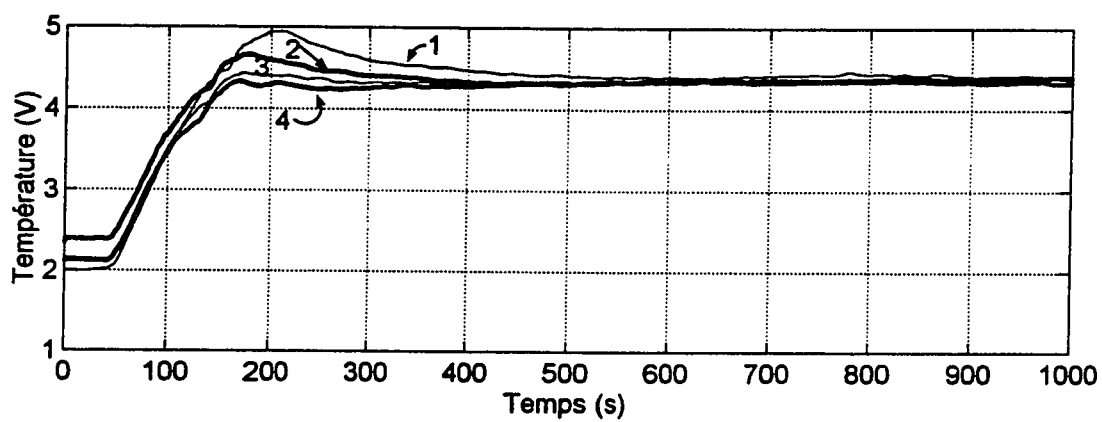


Figure 4.13: Évolution de la réponse en température lors de la première étape du réglage

• Test de la robustesse

On présente le comportement du niveau et de la température face à une perturbation de 1V ajoutée sur la commande de la vanne d'eau froide à $t=1000s$.

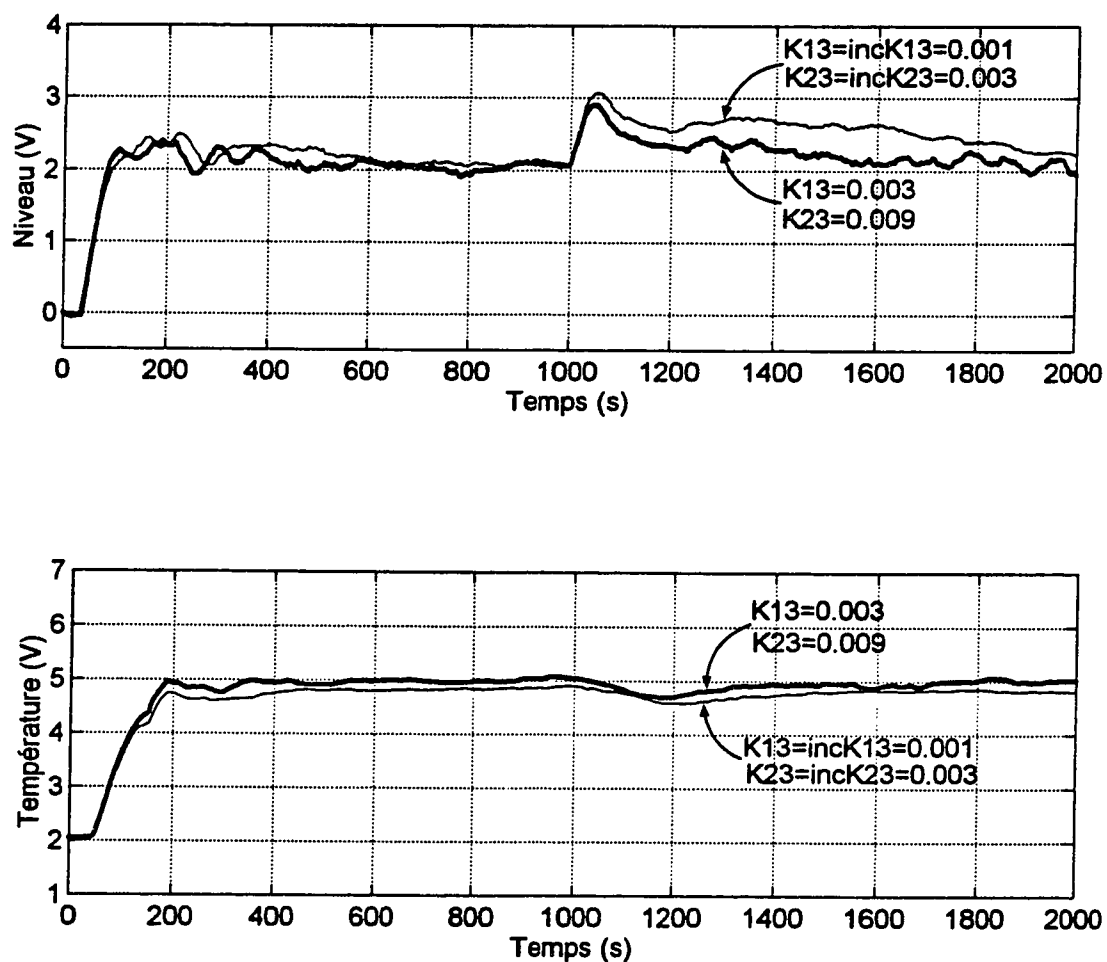


Figure 4.14: Évolution de la robustesse avec l'augmentation de $K13$ et $K23$ (niveau et température)

•Analyse des résultats

Le résultat obtenu en réglant automatiquement et simultanément les compensateurs des deux boucles est meilleur que le résultat obtenu en utilisant simultanément les réglages trouvés séparément et manuellement pour chacune des deux boucles (Cornieles et Bougeret, 1997). De même, si l'on compare le résultat du réglage simultané des deux boucles avec le résultat obtenu en utilisant les réglages trouvés pour chaque boucle (figure 4.15, avec retard et perturbation au milieu de l'expérience), on se rend compte que là encore le réglage multivariable est supérieur.

D'après le tableau 4.4, il semble plus facile de régler la boucle de température. Cela signifie que l'asservissement de température perturbe plus le niveau que l'asservissement du niveau ne perturbe la température. Cela semble normal car les proportions $\frac{1}{1 + \alpha_r}$ et $\frac{\alpha_r}{1 + \alpha_r}$ de la répartition (figure 4.1) sont telles que l'eau ajoutée (ou retranchée) est environ à 5V (en termes de mesure de température), ce qui aide la boucle de température.

Comme le montre la figure 4.14, plus les valeurs de K13 et K23 sont élevées, plus le système converge rapidement vers les consignes de niveau et de température en cas de suivi de consigne ou de régulation.

4.4 Difficultés techniques

Diverses difficultés techniques ont été rencontrées au cours de cette étape notamment au niveau de l'ordinateur dont la mémoire était insuffisante. Le PC a donc dû être changé afin de pouvoir enregistrer et afficher des expériences de plusieurs heures. On a utilisé un Pentium® 166 Mhz avec 32 Mo de mémoire vive.

D'autre part, un filtre du premier ordre avec une constante de temps de 20s a été ajouté sur la mesure de la température. Cela a permis de "lisser" la réponse fournie par le thermocouple et d'introduire un détecteur d'oscillations fonctionnant correctement. La détermination des nombreuses constantes présentes dans les sous-programmes (coefficients μ et μ_k , taux maximal d'ondulation, seuil de détection d'un maximum local, temps de remise à zéro, critère de régime stationnaire atteint...) a également été assez difficile et a demandé beaucoup de tests (notamment nocturnes, pour gagner du temps). En effet le système n'est jamais le même car:

- la pression d'alimentation en eau varie au cours de la journée (hausse de pression le midi par exemple);
- la température de l'eau chaude varie aussi (cycle à 120°F de 8 heures à 16 heures et à 97°F la nuit, cycle différent en fin de semaine, pannes de chauffage, ...);
- la température de l'eau froide a augmenté au printemps puis, bizarrement, est redescendue pendant les jours plus chauds, probablement à cause d'un

système de réfrigération. Mais pendant l'été la température de l'eau froide a remonté.

Enfin, la longueur des expériences a obligé à respecter certains horaires pour lancer les expériences car il est important que la dynamique du système ne varie pas trop au cours de la phase de réglage automatique.

4.5 Conclusions générales sur les expériences

Le système de réglage automatique des gains d'un PID Dual Loop que nous avons mis au point semble bien répondre à la fonction qui lui incombe. Tout comme les autres systèmes de réglage proposés par la littérature, notre système nécessite la détermination préalable d'un certain nombre de paramètres tels que le pas d'incrément des gains, l'amplitude du premier dépassement et le facteur d'amortissement. Cette détermination peut être effectuée en fonction de la classe de systèmes à asservir ainsi que des spécifications d'opérations requises. Dans son application à l'asservissement du niveau et de la température d'un réservoir d'alimentation, notre système a permis un réglage de gains satisfaisant dans un grand nombre d'essais. Ces gains ont fait que le comportement du niveau et de la température converge rapidement et d'une façon stable vers les consignes. Cela, malgré la diversité des conditions de fonctionnement du banc d'essai (variation de la pression d'air, de la température de l'eau d'alimentation, variation du débit de sortie du réservoir et la présence de perturbations électriques).

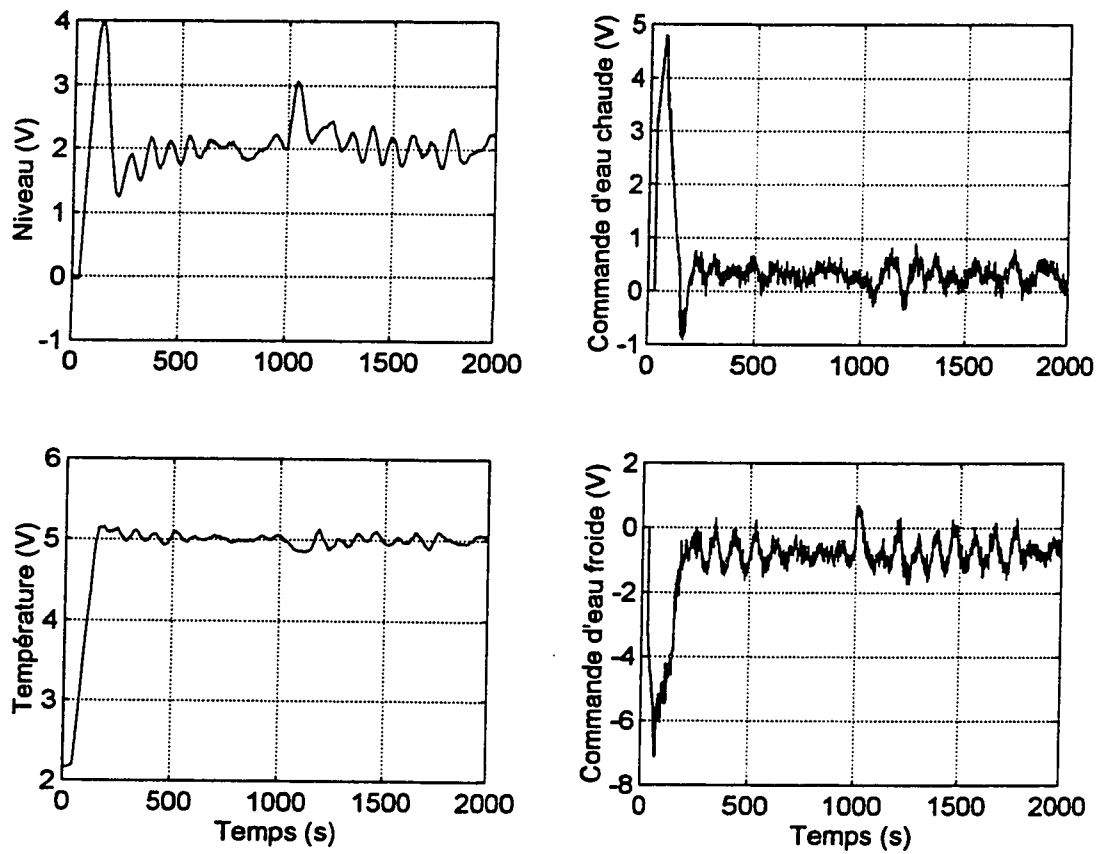


Figure 4.15: Comportement du système lorsque $K_{11}=0.85$, $K_{12}=12.2$, $K_{13}=0.006$, $A_{11}=37.3$, $A_{12}=347$, $K_{21}=2.42$, $K_{22}=35$, $K_{23}=0.05$, $A_{21}=75$, $A_{22}=1416$ (avec un retard de 30s et une perturbation d'un volt sur la commande d'eau froide)

CHAPITRE V

ASSERVISSEMENT DU NIVEAU ET DE LA TEMPÉRATURE AU MOYEN D'UN COMPENSATEUR NEURONAL

5.1 Introduction

L'objectif de ce chapitre est d'illustrer le comportement d'un asservissement du niveau et de la température de notre banc d'essai effectué par voie d'un compensateur neuronal. Les résultats sont ensuite comparés avec le comportement lors de l'emploi du compensateur PID Dual Loop avec le module d'adaptation.

5.2 Le compensateur neuronal

Le compensateur neuronal que nous avons mis en place est décrit en détail à la référence Ruiz et Cornieles (1997). Dans ce qui suit, nous nous limiterons à en présenter les éléments les plus importants.

La structure du compensateur est représentée à la figure 5.1. Les symboles qui interviennent dans cette figure ont la signification suivante:

$y(t)$: variable du procédé à l'instant d'échantillonnage t ; selon la nature spécifique de notre expérience, cette variable représente la sortie du capteur du niveau ou de la température, ou des capteurs du niveau et de la température (cas multivariable).

$u(t)$: variable de contrôle du procédé à l'instant d'échantillonnage t ; elle correspond aux tensions de contrôle fournies par notre compensateur neuronal, $U1$, $U2$ ou $U1$ et $U2$ (encore une fois selon la nature de l'expérience).

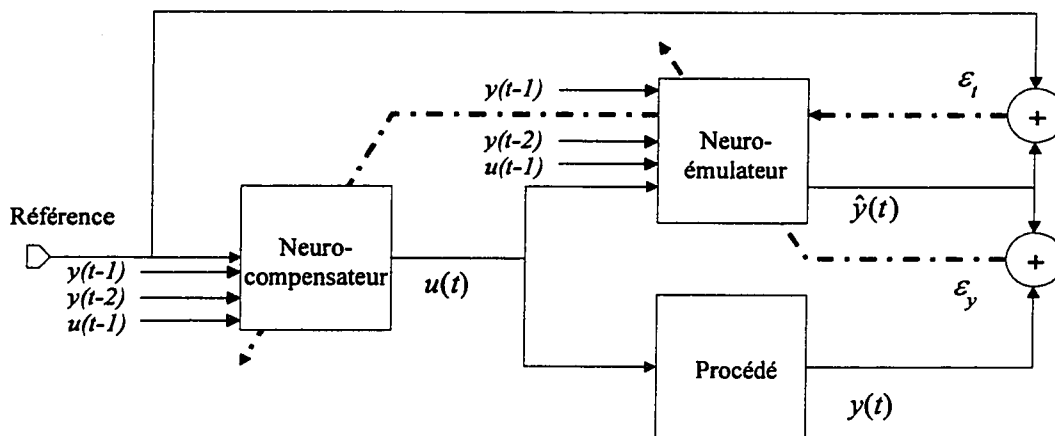


Figure 5.1: Structure "Indirect Learning Architecture"

5.3 Structure du compensateur neuronal

Tel que suggéré par la figure 5.1, le compensateur neuronal est constitué d'un neuro-compensateur et d'un neuro-émulateur. Ces deux éléments sont représentés aux figures 5.2 et 5.4 et sont constitués respectivement de 3 couches (neuro-compensateur) et 2 couches (neuro-émulateur). Les entrées des matrices W_i associés avec ces couches (figure 5.5) correspondent aux gains utilisés pour pondérer les coefficients (poids) du vecteur d'entrée de la couche. Les dimensions de ces matrices dépendent de la nature de l'expérience effectuée. Dans le cas d'un asservissement du niveau et de la température, les dimensions deviennent: $W1=[10 \times 4]$; $W2=[5 \times 10]$; $W3=[1 \times 5]$; $W4=[15 \times 4]$ et $W5=[1 \times 15]$.

5.3.1 Le neuro-compensateur

Le principe est le suivant: on fournit au procédé une entrée $u(t)$ qui produit une sortie $y(t)$. La sortie à son tour est appliquée au neuro-compensateur qui produit alors une sortie $\hat{u}(t)$. Pour le neuro-compensateur, $y(t)$ représente une consigne et $u(t)$ est la commande permettant d'obtenir cette consigne. L'erreur (différence entre u et \hat{u}) est rétro-propagée à travers le réseau. Cette structure est bien connue (Psaltis, Athanasios et Yamamura 1988; Widrow et Nguyen, 1990; Omatu, Khalid et Yusof, 1996) et montrée à la figure 5.2.

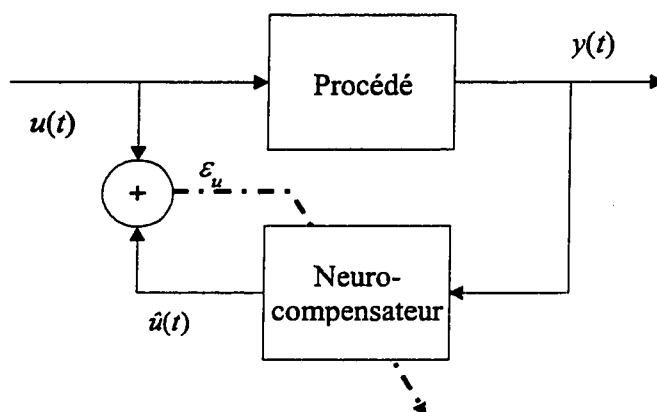


Figure 5.2: Neuro-compensateur. Structure d'apprentissage

5.3.2 Apprentissage "hors ligne"

Les travaux préliminaires ont permis d'obtenir les caractéristiques en boucle ouverte du banc d'essai pour le niveau et la température. De cette façon, on a fait des expériences permettant la génération du tableau entrée/sortie du procédé en boucle ouverte et l'enregistrement de ses valeurs dans un fichier (Ruiz et Cornieles, 1997). À partir de ce fichier, une programmation en Matlab[®] faisant appel aux fonctions ".m" déjà existantes dans le toolbox Neural Networks, a permis l'apprentissage hors ligne d'un réseau de taille variable où le nombre d'entrées est aussi variable. La figure 5.3 montre la génération du fichier entrée/sortie.

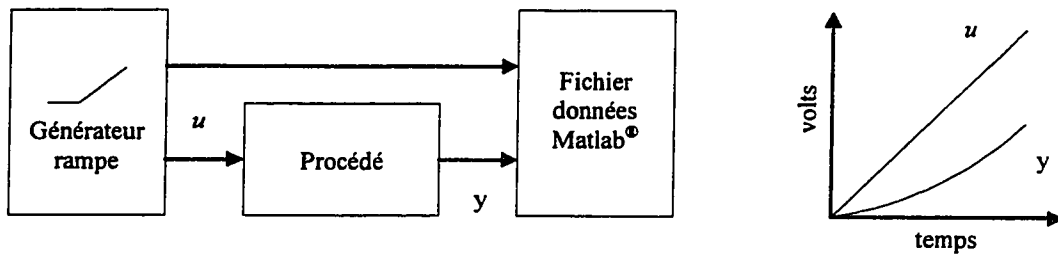


Figure 5.3: Génération du fichier entrée/sortie

C'est ainsi qu'un fichier entrée/sortie a été généré en appliquant une entrée de type rampe sur le procédé afin de connaître les performances de la configuration proposée.

Le choix d'une entrée type rampe réside dans le fait qu'on désire obtenir des couples entrée/sortie "dynamiques", où les valeurs de l'entrée variaient pour couvrir la plage des valeurs de fonctionnement.

À partir du fichier entrée/sortie, quelques points (entre 20 et 30) ont été choisis en favorisant les plus significatifs pour générer un "ensemble d'apprentissage". Avec cet ensemble et en utilisant la fonction "*trainbpx*" du toolbox Neural Networks de Matlab®, l'apprentissage hors ligne de plusieurs réseaux de différentes tailles a été réalisé jusqu'à l'obtention des erreurs quadratiques totales, pour l'ensemble d'apprentissage, acceptables.

Les réseaux ainsi appris étaient censés réaliser la dynamique inverse du procédé. Malheureusement, leurs performances ont été jugées insatisfaisantes indépendamment de la taille du réseau utilisé. L'apprentissage en ligne se montre donc nécessaire.

5.3.3 Apprentissage en ligne

La structure choisie pour l'apprentissage en ligne est bien connue et s'appelle dans la littérature "Indirect Adaptive Control" (Psaltis, Athanasios et Yamamura, 1988; Werbos, 1990; Omatu, Khalid et Yusof, 1996). Étant donné que la dérivée de l'erreur de sortie par rapport à la commande appliquée (jacobien du procédé) n'est pas disponible afin d'appliquer l'algorithme de rétro-propagation (*backpropagation*), il est question d'évaluer cette "sensibilité" de la sortie du procédé par rapport à l'entrée d'une façon artificielle. Un deuxième réseau, connu comme neuro-émulateur, est ajouté au schéma déjà présenté.

L'objectif de ce nouvel élément, copie du procédé, est de permettre la rétro-propagation de l'erreur (référence-sortie) afin d'actualiser les poids du compensateur. La construction du neuro-émulateur est similaire à celle du neuro-compensateur et le schéma de base est montré dans la figure suivante:

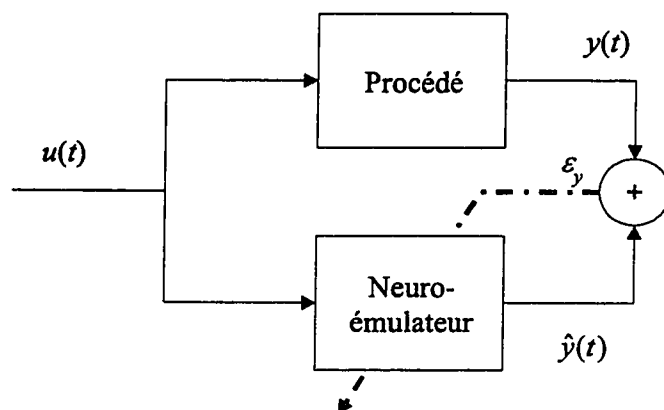


Figure 5.4: Neuro-émulateur. Structure d'apprentissage

Dans un premier temps, on réalise l'apprentissage hors ligne du compensateur et de l'émulateur à l'aide des routines d'apprentissage déjà construites. Après en ligne, les poids du compensateur et de l'émulateur sont actualisés afin de minimiser respectivement l'erreur d'apprentissage $réf(t) - \hat{y}(t)$ et l'erreur d'émulation $y(t) - \hat{y}(t)$.

Les fonctions d'activation choisies sont:

- Tangente hyperbolique (bornes -1 à 1) pour les couches d'entrée et cachées.
- Linéaire pour les couches de sortie.

Les données précédentes des fichiers entrée/sortie, ainsi que les valeurs lues par la carte d'acquisition lors des travaux sur le réservoir, sont normalisées dans l'intervalle -1 à 1 .

5.3.4 Choix des entrées

Afin d'optimiser la taille des réseaux à implanter, on a limité au début des expériences le nombre de couches à 3 (1 couche cachée) et le nombre de neurones par couche à 15. Après l'apprentissage hors ligne et la validation des résultats, ces paramètres se sont révélés suffisants et on a même réduit la taille de l'émulateur à 2 couches (Ruiz et Cornieles, 1997).

Si les poids obtenus pour chaque neurone d'une même couche se ressemblaient entre eux, on réduisait encore le nombre de neurones dans cette couche.

L'intérêt de réduire au maximum la taille des réseaux est évident: étant donné une période d'échantillonnage fixée à 0,5s, le temps disponible pour l'actualisation des réseaux et pour la génération de la commande est donc aussi limité. L'algorithme d'apprentissage "on-line" devrait tenir compte de cette contrainte.

5.3.5 Algorithme d'apprentissage en ligne

Basé sur l'approche d'Omatu, Khalid et Yusof (1996), on a programmé sur Matlab® une fonction (m.m) qui reçoit, comme paramètres de référence, la sortie du système ainsi que ce même signal retardé, et génère la commande pour le procédé.

Cette routine peut appeler un tableau Simulink® ou bien n'importe quel programme Matlab®, ce qui va permettre sa réutilisation lors des expériences et sur le réservoir réel.

L'algorithme d'apprentissage en ligne utilise les poids des réseaux du neuro-compensateur et du neuro-émulateur qui ont déjà été déterminés hors-ligne et enregistrés dans le fichier weights.mat et les modifie à chaque échantillon avant de les ré-enregistrer.

5.4 Algorithme de "Back-propagation"

Comme approche pour l'apprentissage en ligne du neuro-compensateur et du neuro-émulateur, on travaille avec la règle Delta (Youji, Hideakis et Hidekatsu, 1991; Omatu, Khalid et Yusof, 1996), sans "momentum". Ainsi, appliquée sur la structure de la figure 5.5.

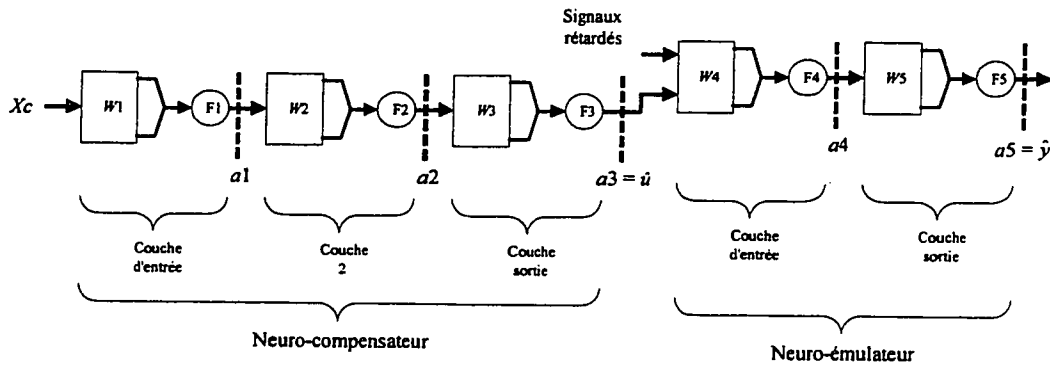


Figure 5.5: Structures du neuro-émulateur et du neuro-compensateur

Le développement de l'algorithme de rétro-propagation de l'erreur donne un ajustement des poids pour chaque couche à partir de l'erreur d'apprentissage (Δw). Les matrices des poids calculées lors des apprentissages hors ligne ont les dimensions suivantes: $W1=[10 \times 4]$; $W2=[5 \times 10]$; $W3=[1 \times 5]$; $W4=[15 \times 4]$ et $W5=[1 \times 15]$. Ainsi, compte tenu la structure présentée à la figure 5.5 où les fonctions d'activation sont linéaires et tangentes hyperboliques pour F5, F3 et F4, F2, F1 respectivement, on obtient:

$$\Delta w^5 = \eta * \varepsilon * a_4 \quad (5.1)$$

$$\delta^4 = [f'(a_4) * \varepsilon] = [1 - (a_4 * a_4)] * \varepsilon \quad (5.2)$$

$$\Delta w^4 = \eta * \delta^4 * a_3 \quad (5.3)$$

$$\delta^3 = [f'(a_3) * (\delta^4 * w^4)] = a_3 * [\delta^4 * w^4]_{1 \times 5} \quad (5.4)$$

$$\Delta w^3 = \eta * \delta^3 * a_2 \quad (5.5)$$

$$\delta^2 = [f'(a_2) * (\delta^3 * w^3)] = [1 - (a_2 * a_2)] * [\delta^3 * w^3]_{1 \times 10} \quad (5.6)$$

$$\Delta w^2 = \eta * \delta^2 * a_1 \quad (5.7)$$

$$\delta^1 = [f'(a_1) * (\delta^2 * w^2)] = [1 - (a_1 * a_1)] * [\delta^2 * w^2]_{1 \times 4} \quad (5.8)$$

$$\Delta w^1 = \eta * \delta^1 * Xc \quad (5.9)$$

où δ^x est l'erreur propagée vers la couche "x", η est le coefficient d'apprentissage et f' est la dérivée de la fonction d'activation correspondante.

En remplaçant " ε " par ε_i ou ε_y , il est possible d'actualiser les poids du neuro-compensateur ou du neuro-émulateur respectivement.

5.5 Résultats expérimentaux

La méthodologie de travail a été:

Construction des réseaux

- Génération de fichier entrée/sortie du procédé (test en boucle ouverte).
- Normalisation des données et sélection d'un nombre de paires réduit (entre 20 et 30 échantillons).
- Apprentissage du neuro-émulateur.
- Validation.
- Optimisation de la taille du réseau si possible.
- Apprentissage du neuro-compensateur.
- Validation.
- Optimisation de la taille du réseau si possible.
- Génération du fichier weights.mat, où sont stockés les poids des réseaux.

Réservoir

Pour le réservoir, on a réalisé une série d'essais en boucle ouverte afin d'identifier la dynamique du système + capteurs + actionneurs. Tout en ignorant la modélisation mathématique du procédé, on s'est servi des expériences en boucle ouverte pour la conception des compensateurs. Système, capteurs, interface électrique et actionneurs sont considérés comme un seul ensemble et, d'après notre approche, une "boîte noire". Pour s'y faire, une rampe est appliquée comme signal de commande. Ainsi, des fichiers dynamiques ont été générés, la valeur de la commande variant d'un point à l'autre.

Trois expériences se sont déroulées: niveau, température, niveau + température.

L'idée est de concevoir un compensateur pour le contrôle du niveau, un deuxième pour la température et, en tenant compte des interactions entre les deux variables, additionner le tout.

Étant donné que le réservoir a déjà été modélisé mathématiquement, plusieurs simulations ont été complétées sur ordinateur avant de travailler sur le vrai réservoir. Cette stratégie a permis d'évaluer les effets des variations des paramètres d'apprentissage sur la performance du contrôle sans avoir à réaliser de longues expériences.

L'aspect le plus intéressant, c'est-à-dire la réalisation d'une commande sur les deux variables, demande la conception d'un réseau linéaire qui prendra comme entrées les commandes générées par les compensateurs de niveau et de température et fournit les commandes pour les vannes d'eau chaude (u_1) et d'eau froide (u_2). Ce nouveau réseau a aussi pour mission le découplage, dans la mesure du possible, entre les deux variables à contrôler.

Différemment des réseaux précédents, les poids du réseau linéaire ne seront pas modifiés lors de son fonctionnement. Les valeurs des poids sont fixées à partir des valeurs du découpleur déjà conçu et implanté lors des travaux précédents sur le réservoir (Cornieles et Bougeret, 1997).

On va étudier ici le développement des tests pour illustrer le comportement de l'algorithme pour la commande de niveau et de température d'un réservoir. Ces tests ont pour but de faire apparaître la capacité du système de contrôle neuronal. Les références sont données en Volts (V). Pour avoir une idée de la variation du niveau, il faut savoir que 2V de variation en sortie du capteur de niveau correspondent à 5cm de variation du niveau dans le réservoir. Toutes les expériences sont réalisées avec un niveau de repos de 0.5m (0V), une référence en niveau de 0.55m (2V) et le robinet de sortie ouvert à une position fixée.

5.5.1 Logiciel du compensateur neuronal

Le compensateur neuronal déjà analysé (apprentissage indépendant de boucles à étudier) est implanté dans le fichier m.m. Il s'agit d'un programme écrit avec le langage commercial Matlab®. La liste du logiciel se trouve en annexe.

Données en entrées modifiables par l'utilisateur:

- Type d'expérience (Niveau, Température, Niveau + Température).
- Introduction d'une perturbation sur le système (Oui/Non).
- Introduction d'un retard sur la commande (Oui/Non et si oui retard exprimé en moitié de seconde).
- α_r (rapport d'eau froide/eau chaude).
- Les références en niveau et température exprimées en Volts.
- La période d'échantillonnage du compensateur.

Le nombre d'échantillons sur lesquels va porter l'expérience. On remarque que le produit de la fréquence d'échantillonnage avec le nombre d'échantillons nous donne la durée de l'expérience.

Toutes ces variables ont une valeur par défaut et l'utilisateur du logiciel n'a pas à initialiser toutes les données à chaque utilisation.

Informations retournées par le logiciel:

Lorsque l'expérience est terminée, le logiciel nous donne l'évolution de la position et de la commande de chaque boucle dans le temps. L'utilisateur peut également sauvegarder ces mêmes données sous forme de fichier utilisable par Matlab[®]. Le logiciel offre aussi la possibilité d'enregistrer les poids des réseaux après l'expérience.

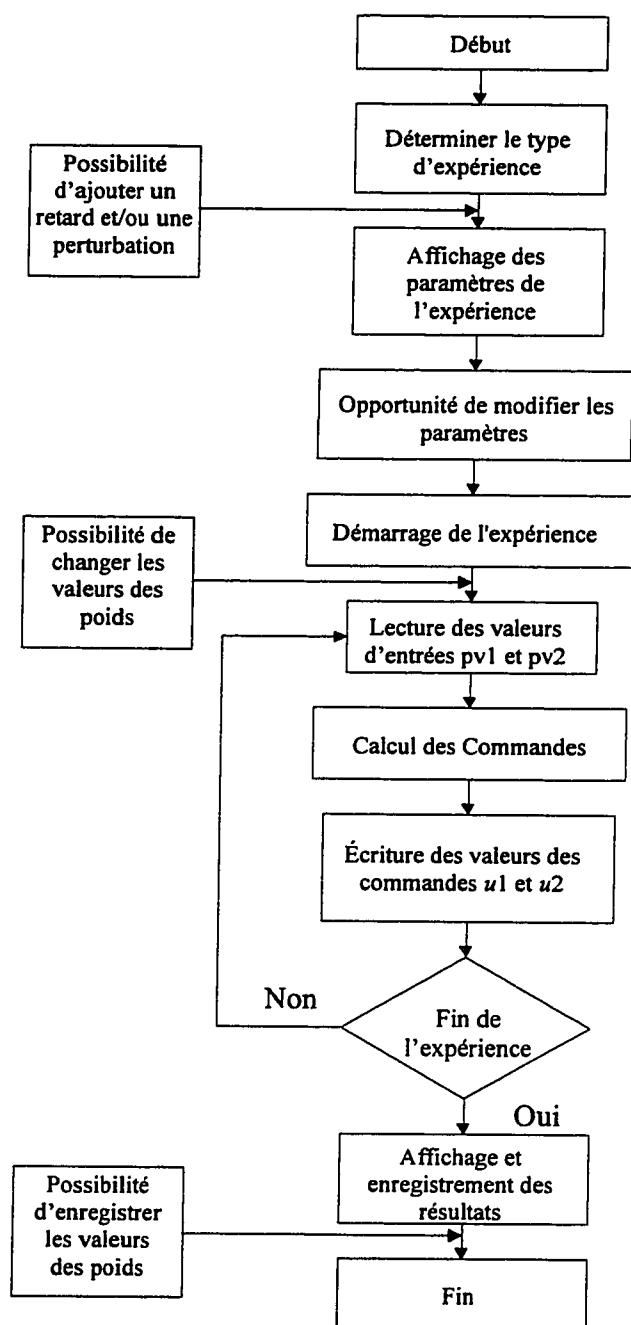


Figure 5.6: Algorithme du logiciel de contrôle

5.5.2 Contrôle du niveau

La commande du niveau générée par le compensateur est nommée $U1$. Elle est appliquée sur le découpleur afin de fournir les deux commandes $u1$ et $u2$, commandes d'eau chaude et d'eau froide respectivement. Le découpleur est, dans ce cas-ci, un facteur d'échelle qui a pour objectif d'économiser la consommation d'eau chaude. Ainsi, le rapport choisi de façon expérimentale étant $\alpha_r = 1.7$ (Cornieles et Bougeret, 1997), les équations des variables $u1$ et $u2$ sont:

$$u1 = \left(1 - \frac{\alpha_r}{1 + \alpha_r}\right) * U1 = \frac{1}{3} * U1 \quad (5.10)$$

$$u2 = \frac{\alpha_r}{1 + \alpha_r} * U1 = \frac{2}{3} * U1 \quad (5.11)$$

Noter que, le système étant réel, il est normal de retrouver des écarts entre les réponses lors des différentes expériences car les variations de la pression du réseau d'air comprimé pour les vannes sont importantes pendant la journée. Le compensateur doit faire face à cette contrainte. L'ensemble des tests proposés sera réalisé avec le même compensateur afin de tester sa robustesse et sa versatilité.

5.5.2.1 Contrôle du niveau seul

Objectif: Mise en évidence des performances de fonctionnement de la régulation de niveau par le contrôle neuronal.

Modalités: Dans cette série de tests, on ne se préoccupe pas à garder la valeur de la température constante. Le robinet de sortie est ouvert au point fixé et on fixe les valeurs de repos des vannes de manière à équilibrer le niveau à 0.5m (0V). Lorsque le système est stable, c'est-à-dire que le niveau est stationnaire, on peut lancer l'expérience. La valeur de consigne est fixée à 2V.

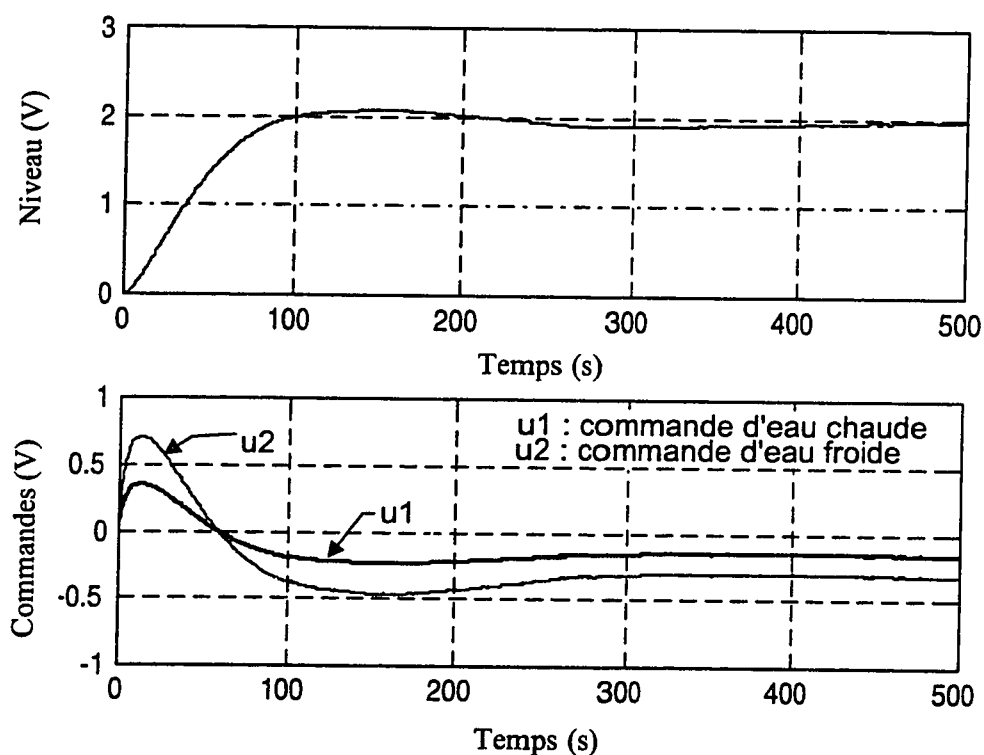


Figure 5.7: Test de régulation du niveau

Résultats et commentaires: Le compensateur offre une bonne performance en rapidité et précision. Il existe un léger dépassement mais l'erreur d'apprentissage est annulée. On peut remarquer quand même des oscillations amorties autour de la valeur finale. Les signaux de commande ne sont pas agités et leurs formes sont identiques mais affectées par le rapport fixe du découpleur. Les résultats sont en général satisfaisants et on constate une bonne performance du compensateur.

5.5.2.2 Contrôle du niveau en présence d'une perturbation

Objectifs: Mise en évidence des performances de fonctionnement de la régulation de niveau pour le contrôle neuronal et sa capacité de réponse face à une perturbation sur le système.

Modalités: Dans les mêmes conditions que l'expérience précédente, une fois le régime permanent assuré (1000s), on applique une perturbation (échelon de 1V d'amplitude) sur le signal de commande de la vanne d'eau froide.

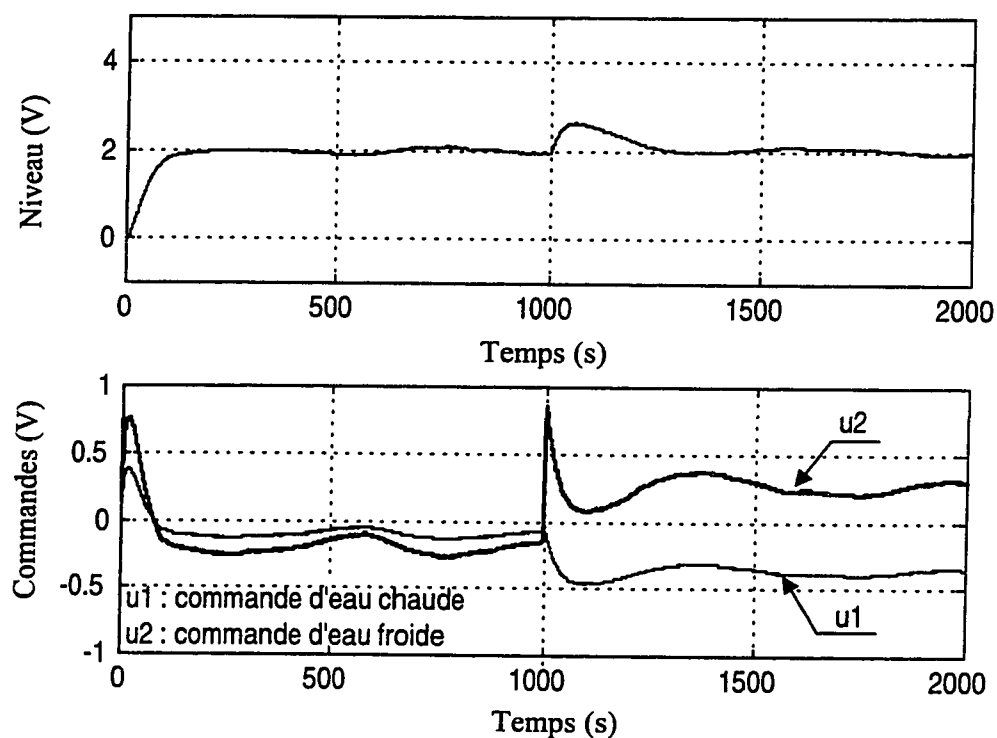


Figure 5.8: Contrôle du niveau en présence d'une perturbation

Résultats et commentaires: La réponse initiale du système est similaire à celle obtenue dans l'expérience précédente. La perturbation introduite provoque une augmentation du débit d'entrée au réservoir. Le compensateur répond très vite en réduisant le débit global d'entrée pour neutraliser la perturbation et récupérer la consigne et la stabilité. L'expérience est satisfaisante.

5.5.2.3 Robustesse du compensateur en présence d'une perturbation et d'un retard

Objectif: Étudier la robustesse du compensateur neuronal sous les conditions d'opération visant à reproduire la réalité industrielle, c'est-à-dire, retard sur la commande et perturbation.

Modalités: Le compensateur utilisé ici est identique à celui utilisé précédemment. Par rapport au cas précédent, un retard de 20s a été introduit sur la commande.

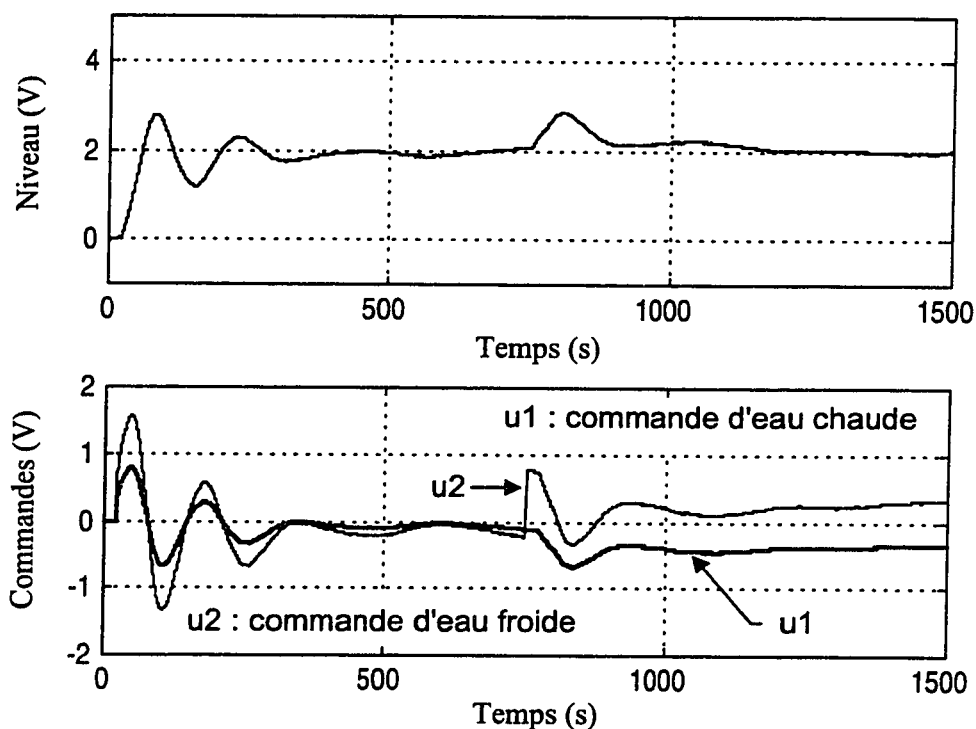


Figure 5.9: Robustesse du compensateur en présence d'une perturbation et d'un retard

Résultats et commentaires: On observe les oscillations typiques d'un système avec retard, mais le compensateur réagit encore bien à la perturbation. On peut conclure cette étape en remarquant, comme aspect positif, la robustesse du compensateur et, comme aspect négatif, la difficulté montrée à maîtriser le retard.

5.5.3 Contrôle de la température

Parallèlement au cas précédent, un compensateur qui aura pour objectif la commande de la température a été conçu. Le compensateur génère le signal de commande $U2$ qui, une fois appliqué sur le découpleur, fournit $u1$ et $u2$, commandes des vannes d'eau chaude et d'eau froide respectivement. La valeur expérimentale du β_1 , coefficient d'eau chaude, est 0.36 et celle du β_2 , coefficient d'eau froide, -0.64; celles-ci ont été trouvées lors des travaux précédents (Cornieles et Bougeret, 1997).

Donc

$$u1 = \beta_1 * U2$$

$$u2 = \beta_2 * U2$$

Il faut remarquer que le système physique possède un retard naturel d'environ 20s dû au temps nécessaire pour le mélange et l'homogénéisation de l'eau arrivant avec toute la masse d'eau déjà stockée dans le réservoir (Cornieles et Bougeret, 1997).

La mesure de la température est assurée par un thermocouple qui présente une sensibilité trop élevée. Un filtre numérique du 1^{ier} ordre avec une constante du temps de 20s a été conçu afin de lisser la réponse de l'ensemble capteur-interface électrique (Cornieles et Bougeret, 1997).

Comme dans le cas du contrôle du niveau, un seul compensateur a été soumis à l'ensemble des expériences.

5.5.3.1 Contrôle de la température seule

Objectif: Illustrer le comportement de la température avec un compensateur neuronal, le niveau étant réglé manuellement à 0.5m (136 litres).

Modalités: Les tests réalisés pour l'asservissement de température sont de même type que ceux réalisés pour l'asservissement de niveau. Pour ce test, on ouvre le robinet de sortie au point fixé puis on choisit le point de repos pour les débits d'eau froide et d'eau chaude de façon à équilibrer le niveau à 0.5m et à fixer la température de repos souhaitée. La consigne désirée étant fixée à 6V, on démarre l'expérience.

Pour s'assurer des bonnes conditions de fonctionnement, on stabilise manuellement le niveau à l'aide du potentiomètre de régulation d'alimentation des vannes d'eau chaude

et d'eau froide.

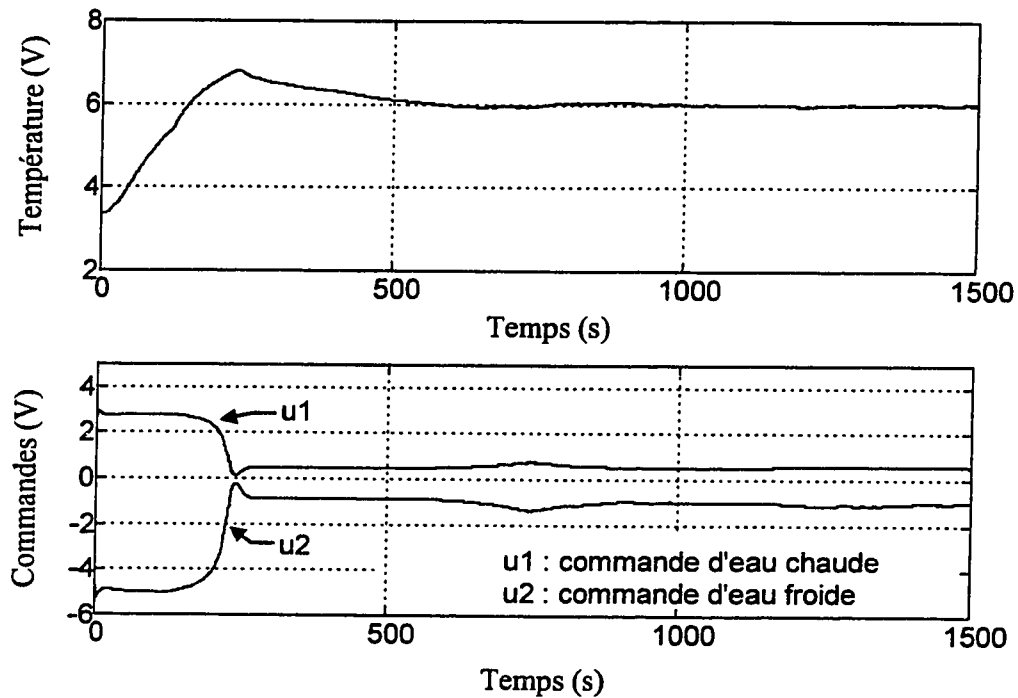


Figure 5.10: Contrôle de la température

Résultats et commentaires: On remarque un très fort dépassement qui, malgré nos efforts, n'a pas pu être réduit. Cette contrainte a comme conséquence une mauvaise performance en rapidité. Le changement de pente dans la première partie de transitoire de la réponse est témoin des variations de la pression de l'eau arrivant au réservoir car, pour une commande de valeur constante, il y a une variation du débit d'entrée.

5.5.3.2 Contrôle de la température en présence d'une perturbation

Objectif: Mise en évidence de la capacité de réponse du compensateur neuronal face à une perturbation sur le système.

Modalités: Dans les mêmes conditions que l'expérience précédente, une fois assuré du régime permanent (1000s), on applique une perturbation artificielle (échelon de $-3V$) sur la commande de température qui se traduit, après l'action du découpleur, en un échelon de $-1V$ sur la commande d'eau chaude et de $2V$ sur la commande d'eau froide.

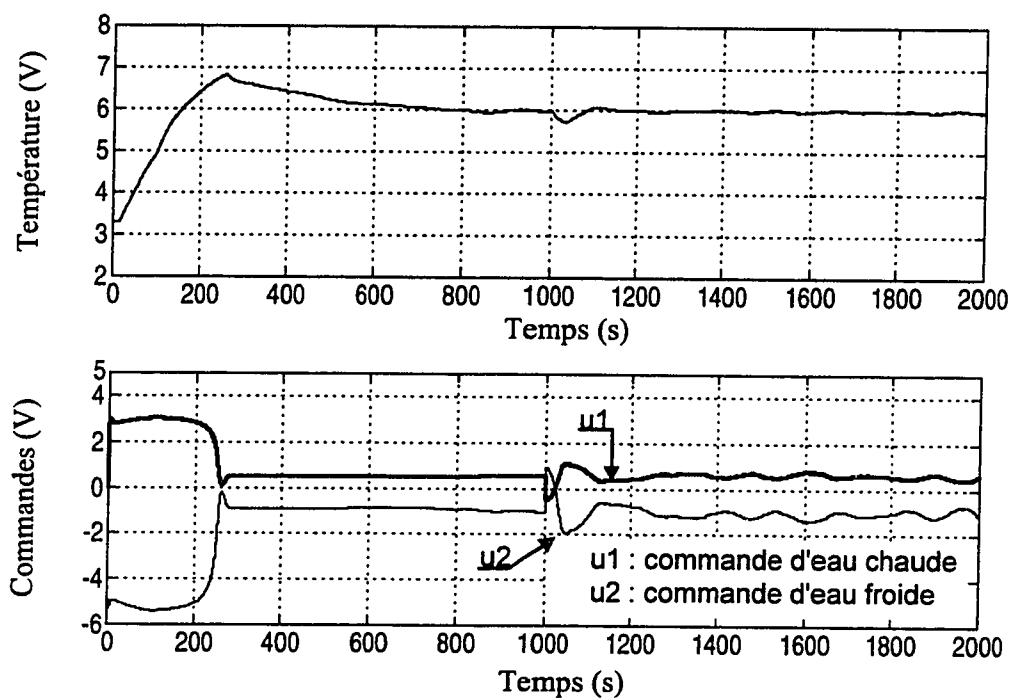


Figure 5.11: Contrôle de la température en présence d'une perturbation

La température du réservoir tendant à diminuer, le compensateur doit réagir pour compenser les nouvelles conditions d'opération.

Résultats et commentaires: Le compensateur offre une bonne réponse en régulation réagissant rapidement et efficacement à la perturbation. On remarque aussi la qualité des signaux de commande.

5.5.3.3 Robustesse du compensateur en présence d'une perturbation et d'un retard

Objectif: Étudier la robustesse du compensateur neuronal sous les conditions d'opération visant à reproduire la réalité industrielle, c'est-à-dire, retard sur la commande et perturbation.

Modalités: Le compensateur utilisé et la perturbation introduite sont identiques à ceux utilisés lors de l'expérience précédente. De plus, au retard naturel du système (20s), on a introduit un retard additionnel de 30s sur la commande.

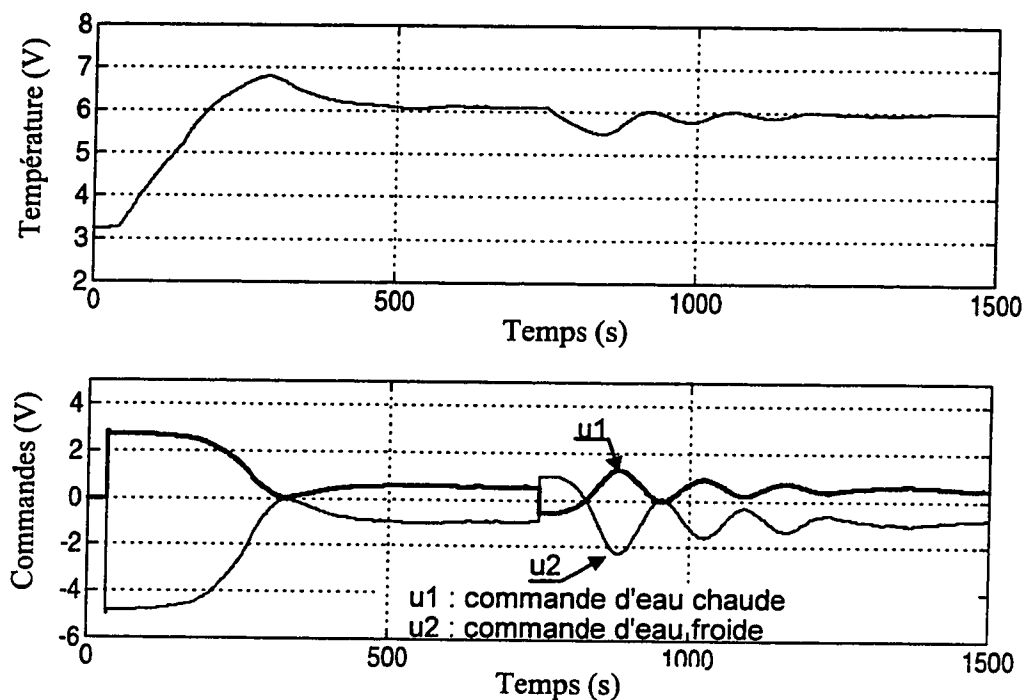


Figure 5.12: Robustesse du compensateur en présence d'une perturbation et d'un retard

Résultats et commentaires: Le retard appliqué ne détériore pas excessivement les performances obtenues lors de l'expérience précédente. On observe les oscillations sur la température dans le processus de récupération après la perturbation. En général, on constate un bon comportement en régulation et une performance moins brillante en asservissement.

5.5.4 Test des deux boucles scalaires

Objectifs: Les expériences suivantes ont pour but d'illustrer le comportement de deux compensateurs conçus séparément mais utilisés simultanément à l'aide du découpleur. Les interactions entre les deux variables à contrôler étant très fortes, cette série d'essais a comme objectif additionnel de tester l'efficacité du découpleur proposé.

Modalités des tests: Les expériences sont réalisées sous les mêmes conditions qu'aux parties 5.5.2 et 5.5.3 (mêmes compensateurs, mêmes paramètres et mêmes consignes) sauf que maintenant, les deux boucles fonctionnent de façon simultanée. Le robinet de sortie est ouvert à son point fixé et on fixe le débit de repos pour les vannes d'eau chaude et d'eau froide de manière à stabiliser le niveau à 0.5m et obtenir la température de repos désirée. Ces réglages ne seront pas modifiés pendant l'expérimentation.

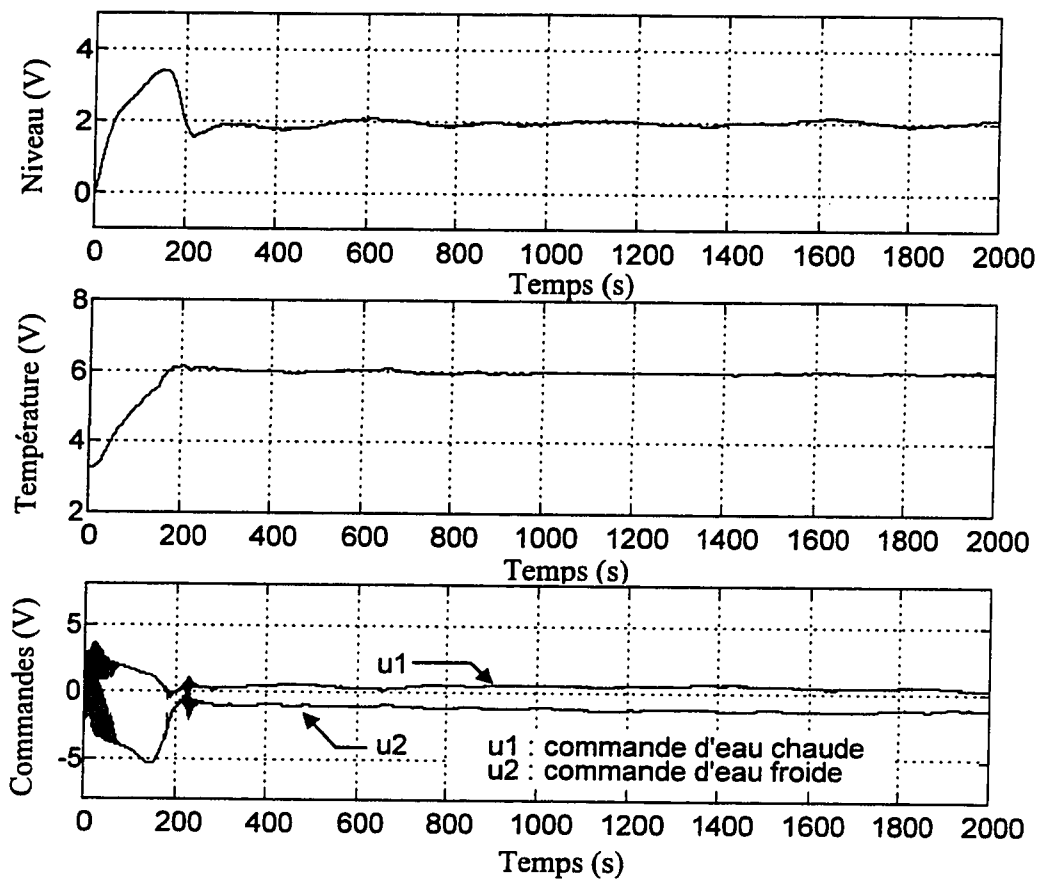


Figure 5.13: Contrôle des deux boucles scalaires

Résultats et commentaires: La stratégie de contrôle développée est correcte: le compensateur ouvre la vanne d'eau chaude et ferme celle d'eau froide pour faire monter le niveau et la température; une fois la consigne de niveau atteint, le débit est réduit afin de minimiser le dépassement du niveau. La régulation de la température fonctionne aussi, le compensateur fixe le débit global (eau froide + eau chaude) pour garder le niveau désiré.

La forte interaction entre les deux variables provoque le dépassement observé dans le niveau. On remarque un excellent contrôle sur la température. Les commandes sont, par contre, très agitées au début de l'expérience pour se stabiliser une fois les consignes rapprochées.

5.5.4.1 Contrôle des deux boucles en présence d'une perturbation

Objectif: Mise en évidence de la capacité de réponse du compensateur neuronal face à une perturbation sur le système.

Modalités: Dans les mêmes conditions que l'expérience précédente, une fois le régime permanent assuré (1000s), on applique une perturbation (échelon de 1V) sur la commande d'eau froide. Deux effets simultanés sont déclenchés: la température du réservoir tendant à diminuer et le niveau du réservoir à augmenter.

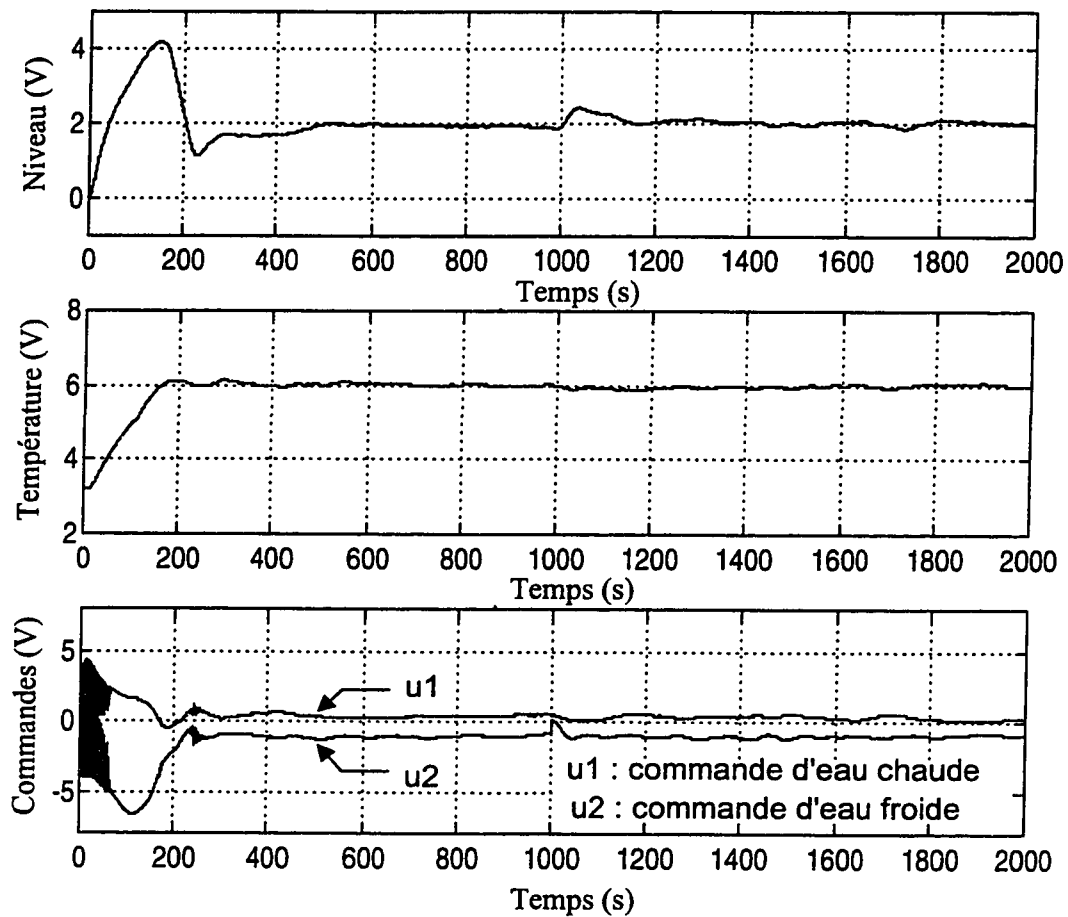


Figure 5.14: Contrôle des deux boucles en présence d'une perturbation

Résultats et commentaires: Le compensateur répond de façon très efficace face à la perturbation, mettant en évidence son bon comportement en régulation. On constate des perturbations électriques au début de l'expérience pour la commande d'eau chaude et la commande d'eau froide. On remarque aussi que la température perturbe le niveau car le dépassement est autour de 100% en régime transitoire.

5.5.4.2 Robustesse du compensateur en présence d'une perturbation et d'un retard

Objectif: Étudier la robustesse du compensateur neuronal sous les conditions d'opération visant à reproduire la réalité industrielle, c'est-à-dire, retard sur la commande et perturbation.

Modalités: Les compensateurs utilisés sont identiques à ceux utilisés lors des expériences de niveau et de température. De plus, au retard naturel du système (20s), on ajoute un retard additionnel de 20s sur la commande. Le robinet de sortie est ouvert à son point fixé et on règle le débit de repos pour les vannes d'eau chaude et d'eau froide de manière à stabiliser le niveau à 0.5m et à obtenir la température de repos désirée. Une perturbation à 1000s (échelon de 1V) est appliquée sur le signal de commande de la vanne d'eau froide.

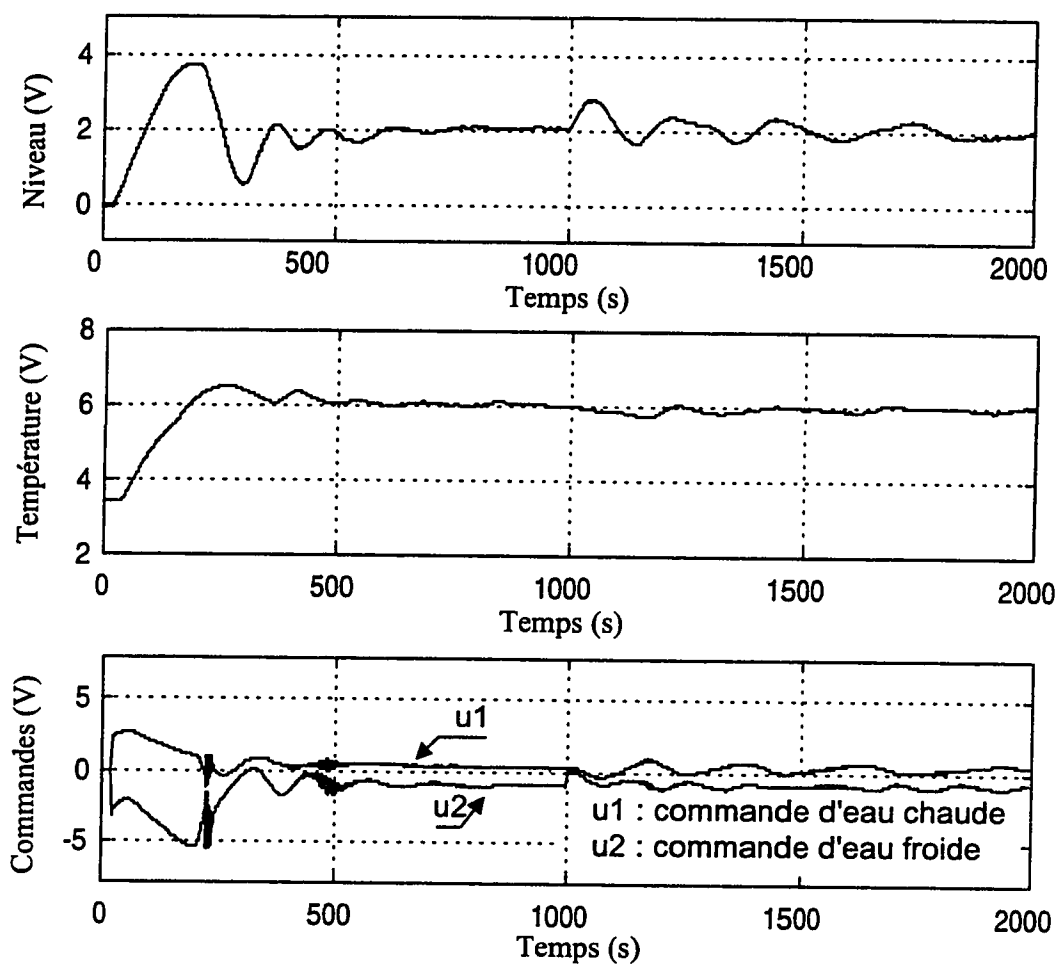


Figure 5.15: Robustesse du compensateur en présence d'une perturbation et d'un retard

Résultats et commentaires: On constate que la réponse du niveau et de la température ont des oscillations types d'un système avec retard. On remarque aussi que le compensateur neuronal réagit bien face à la perturbation ajoutée sur la commande d'eau froide.

5.6 Conclusion sur les expériences

L'analyse des résultats obtenus peut être résumée de la façon suivante:

Pour le contrôle des boucles indépendantes, les compensateurs offrent un fonctionnement satisfaisant. Bien que leurs performances ne soient pas nécessairement spectaculaires en ce qui concerne la rapidité de réponse, la robustesse et l'immunité aux perturbations des compensateurs nous paraissent d'un niveau raisonnable. D'autre part, l'introduction d'un retard sur la commande provoque l'apparition d'oscillations importantes.

Pour le contrôle simultané des deux boucles, les performances sont supérieures à celles qui caractérisent le comportement lors de l'utilisation d'une seule boucle.

Les difficultés de conception associées avec le dimensionnement des différents éléments du compensateur (choix de la taille: nombre de couches et de neurones), peuvent être résolues d'une façon satisfaisante par voie expérimentale sur le banc d'essai.

CHAPITRE VI

COMPARAISON EXPÉRIMENTALE D'UN COMPENSATEUR NEURONAL ET D'UN COMPENSATEUR PID DUAL LOOP

6.1 Introduction

L'objectif de ce chapitre est de comparer les résultats relatifs au module d'adaptation de réglage pour le compensateur PID Dual Loop présenté au chapitre IV et les résultats relatifs au compensateur neuronal présenté au chapitre V. L'étude expérimentale est menée sur le banc d'essai décrit au chapitre II où on asservit le niveau et la température de l'eau dans un réservoir (voir photo en annexe 1).

6.2 Résultats expérimentaux

On présente maintenant les tests réalisés afin de comparer les performances des deux compensateurs mentionnés ci-dessus. Les tests ont été réalisés avec les mêmes conditions d'opérations. On doit se rappeler que K11, K12, K13, K14, A11, A12 sont les paramètres du compensateur PID Dual Loop pour le niveau et K21, K22, K23, K24, A21, A22 les paramètres du compensateur de température. Ces valeurs ont été présentées au chapitre V. Les références sont données en volts. Pour avoir une idée de la variation du niveau, il faut savoir que 2V de variation en sortie du capteur de pression correspondent à 5 centimètres de variation du niveau dans le réservoir. Toutes les

expériences pour les deux compensateurs sont réalisées avec un niveau de repos de 0.5m (0V), une consigne en niveau de 0.55m (2V), une température de repos de 23°C (3,25V), une consigne en température de 35°C (6V) et le robinet de sortie ouvert à une position fixée. Les expériences ont été réalisées à la fin de l'été.

6.3 Logiciel d'asservissement du niveau et de la température

L'asservissement est réalisé par le logiciel Pint_edu.m pour le compensateur PID Dual Loop et m.m pour le compensateur neuronal. Il s'agit de sous-programmes en C. L'asservissement est constitué par la structure présentée au chapitre III pour le compensateur PID Dual Loop et celle présentée au chapitre V pour le compensateur neuronal.

Dans les deux programmes l'utilisateur peut modifier pour chaque test:

Les paramètres des PID Dual Loop des deux boucles.

Les consignes réf1 et réf2 en niveau et en température exprimées en volts.

La période d'échantillonnage du compensateur.

Le choix de perturber le système:

- avec un échelon de 1V ajouté sur la commande de la vanne d'eau froide au milieu de l'expérience (utilisé lors du test de robustesse de l'asservissement de niveau seul et lors du test de robustesse de l'asservissement simultané des deux boucles scalaires);

- avec un échelon de $-0.98V$ ($-0.98 = 3 \cdot \beta_1$) sur la commande de la vanne d'eau chaude et un échelon de $1.92V$ ($1.92 = 3 \cdot \beta_2$) sur la commande de la vanne d'eau froide (utilisés lors du test de la robustesse de l'asservissement de température seule afin de perturber le moins possible le niveau).

Le choix d'un retard sur la commande.

6.4 Test sur le contrôle du niveau seul

6.4.1 Asservissement du niveau seul

Objectif: Mise en évidence des différences de performances de l'asservissement de niveau en présence d'un retard et d'une perturbation sur la commande pour les compensateurs neuronal et PID Dual Loop.

Modalités: On utilise les mêmes modalités présentées au chapitre III pour l'asservissement du niveau seul avec le compensateur PID Dual Loop. Grâce aux programmes développés, on introduit un retard de 20s et une perturbation sur la commande de la vanne d'eau froide. Pour le compensateur neuronal, on introduit les poids déterminés au chapitre V. Pour chaque compensateur, on fait l'expérience du niveau seul, ensuite on fait la superposition des réponses obtenues pour comparer et

analyser les résultats.

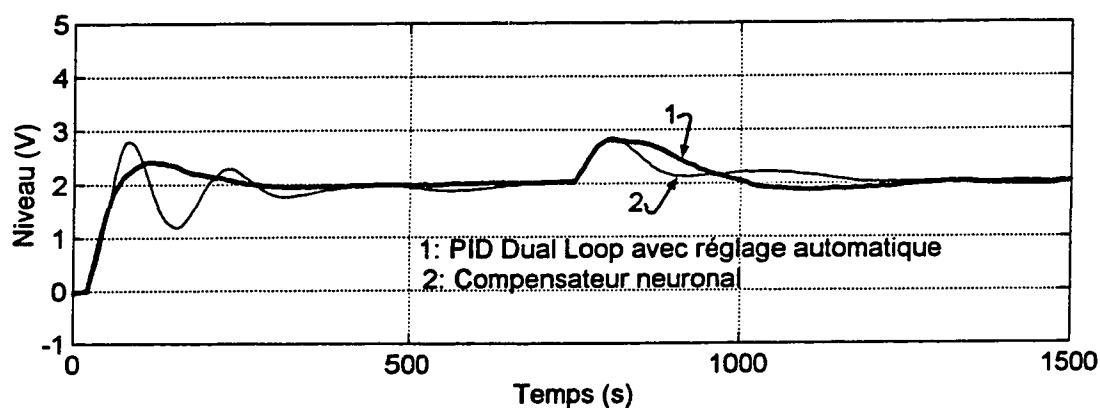


Figure 6.1: Régulation de niveau. Comparaison de deux compensateurs

Résultats et commentaires: Le contrôleur neuronal donne une réponse avec un dépassement voisin de 45% et des oscillations très marquées en régime transitoire. La réponse face à une perturbation est satisfaisante. Avec le compensateur PID Dual Loop, il n'y a pas d'oscillations. Le dépassement est beaucoup plus faible (environ 20%) et le temps de réponse est l'ordre de 250s.

6.5 Test sur le contrôle de la température seule

6.5.1 Asservissement de la température seule

Objectif: Mise en évidence des performances de l'asservissement de température en présence d'un retard et d'une perturbation sur la commande pour les compensateurs neuronal et PID Dual Loop.

Modalités: On utilise les mêmes modalités que pour l'asservissement de niveau. De plus, au retard naturel du système (20s), on introduit un retard additionnel de 30s sur la commande et une perturbation à 750s. Cette perturbation de -0.98V sur la vanne d'eau chaude et de 1.92V sur la vanne d'eau froide est celle décrit au paragraphe 6.3.

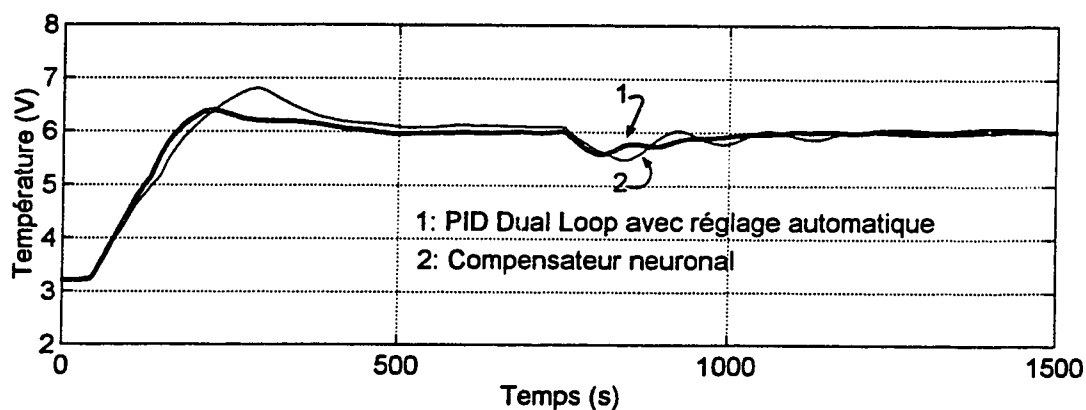


Figure 6.2: Comparaison de la réponse de la température avec deux compensateurs

Résultats et commentaires: Le PID Dual Loop réagit mieux que le compensateur neuronal en régime transitoire et face à la perturbation car il diminue le dépassement qu'elle provoque. Le régime permanent est rejoint en 500s environ. Ce test de robustesse est assez significatif puisque la perturbation introduite est importante.

6.6 Test des deux boucles scalaires

6.6.1 Asservissement simultané du niveau et de la température

Objectifs: Illustrer le comportement des deux compensateurs pour l'asservissement du niveau et de la température. Mettre en évidence les performances en présence d'un retard et d'une perturbation.

Modalités: Les expériences sont réalisées sous les mêmes conditions qu'aux parties 6.4 et 6.5 sauf que maintenant les deux boucles fonctionnent de façon simultanée. On conserve les valeurs de gains obtenues pour les régulations séparées (cas PID Dual Loop) et les poids pour le cas du compensateur neuronal. On introduit un retard de 20s sur les commandes. Ce retard agit sur la boucle du niveau et sur la boucle de la température. Le robinet de sortie est ouvert à son point fixé et on fixe le débit de repos pour les vannes d'eau chaude et d'eau froide de manière à stabiliser le niveau à 0.5m (0V) et obtenir la température de repos désirée 23°C(3.25V). On utilise la structure du découpleur présentée aux chapitres III et IV.

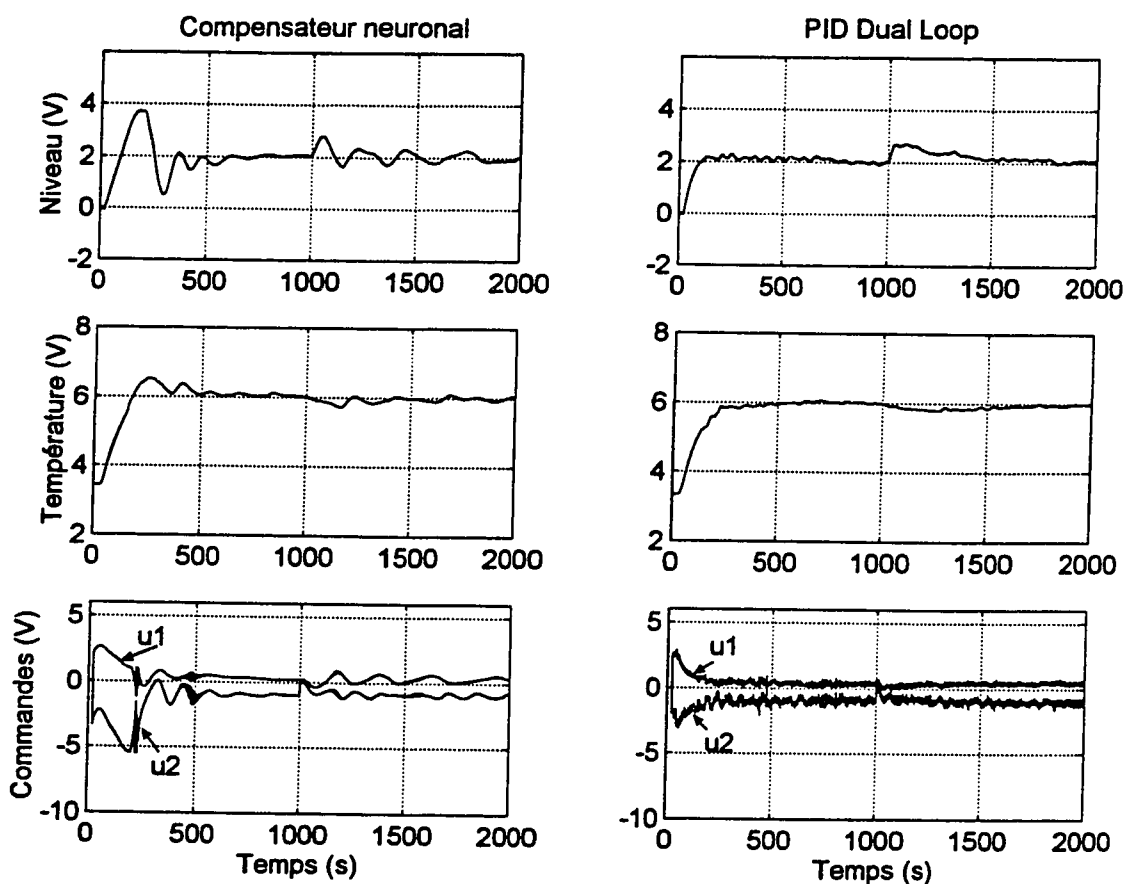


Figure 6.3. Comparaison de la réponse du niveau et de la température avec les deux compensateurs

Résultats et commentaires: Le contrôleur neuronal provoque des oscillations remarquables avec un dépassement d'environ 90% pour le niveau. La réponse de la température est satisfaisante. Le compensateur PID Dual Loop permet d'atteindre les consignes de niveau et de température en un temps raisonnable. On s'aperçoit que les performances en suivi de consigne sont supérieures pour le compensateur PID Dual

Loop. On remarque aussi que le module de réglage automatique pour chaque boucle de régulation donne un réglage très satisfaisant. En effet, la perturbation n'engendre qu'un dépassement de 40% pour le PID Dual Loop. De plus, le PID Dual Loop permet de rejoindre beaucoup plus rapidement le régime stationnaire (en 500s).

6.7 Conclusion générale sur les expériences

L'ensemble des expériences réalisées est très satisfaisant et on constate que les performances en suivi de consigne sont supérieures pour le compensateur PID Dual Loop (avec module de réglage automatique). Également, la robustesse du PID Dual Loop est supérieure dans le cas de la régulation de température et pour le cas multivariable. On confirme que le module de réglage automatique des gains d'un PID Dual Loop donne des résultats très satisfaisants.

CONCLUSION GÉNÉRALE

Notre étude comparative des procédures de réglage des gains de la nouvelle classe de compensateurs PID Dual Loop avec les procédures de réglage actuellement utilisées dans l'industrie nous a permis de mettre en perspective les modalités de réglage des gains d'un PID Dual Loop avec celles associées aux méthodes de Ziegler-Nichols, Rivera-Morari, Cohen-Coon et ITAE. Elle nous a permis également de mettre en perspective la mise en opération d'un compensateur PID Dual Loop avec celle d'un compensateur neuronique. Bien qu'il soit toujours imprudent d'établir des conclusions définitives à partir d'étude de cette nature, nos résultats indiquent que le comportement dynamique d'un procédé de type industriel avec la méthode PID Dual Loop est en général supérieur à celui obtenu avec les autres méthodes.

Cette supériorité est surtout notée en ce qui concerne les méthodes de Ziegler-Nichols et de Cohen-Coon. D'autre part, bien que la méthode de réglage du PID Dual Loop soit peu exigeante au point de vue de la difficulté et du temps requis pour son exécution, il reste que les méthodes de Ziegler-Nichols et de Cohen-Coon sont encore moins exigeantes. D'ailleurs, les méthodes de Rivera-Morari et ITAE sont d'un niveau de complexité équivalent à celui du PID Dual Loop, alors que la mise en opération d'un compensateur neuronal est considérablement plus complexe. En particulier, des difficultés de conception ont été rencontrées au niveau du dimensionnement des différents éléments du compensateur neuronal (choix de taille: nombre de couches et de

neurones). L'algorithme d'apprentissage en ligne a aussi présenté des difficultés de conception et d'implantation. Ces difficultés ont pu être surmontées en procédant par tâtonnement.

Ces considérations sont basées sur des expériences effectuées sur un banc d'essai de type industriel que nous avons mis en place. Ce banc d'essai est constitué d'un asservissement en niveau et en température d'un réservoir d'alimentation en eau et présente les aspects caractéristiques d'un procédé industriel. Ces aspects concernent la structure du matériel et du logiciel, les capteurs et les vannes de contrôle, l'adaptation des filtres de traitement de signaux, la présence de perturbations, le comportement non linéaire, les phénomènes électriques, etc. Son utilisation implique la mise au point d'un grand nombre de détails techniques, notamment ceux reliés aux interfaces électriques et logicielles (horloge temps réel, lecture et écriture sur la carte Labmaster[®], identification des composantes hydrauliques). Il représente un bon exemple de système non stationnaire car il n'était jamais le même: la pression d'alimentation de l'eau variait au cours de la journée, la température de l'eau chaude variait aussi de façon importante. Il s'agit donc d'un banc d'essai bien approprié à émuler un procédé industriel qui, d'ailleurs, a déjà servi à cet effet à au moins une demi-douzaine de chercheurs de l'École Polytechnique.

Le module de réglage automatique des gains d'un PID Dual Loop que nous avons mis en place s'est révélé fiable et efficace. Bien que ce module soit à l'heure actuelle limité

à une fonction d'autoréglage lors de la mise en opération des contrôleurs, il n'est pas difficile d'envisager des modifications afin d'étendre cette fonction à celle d'un compensateur adaptatif en temps réel qui resterait actif durant toute la phase d'opération de l'asservissement du procédé.

En conclusion, ce projet de recherche a contribué à développer de nouvelles techniques et des algorithmes de réglage assistés par ordinateur pour le compensateur PID Dual Loop. Cette contribution a donné la possibilité d'améliorer des techniques qui existent déjà et d'optimiser les procédures de réglage du compensateur PID ayant pour but de réduire les coûts engendrés par celui-ci pour un compensateur sur un site industriel, de réduire également le délai de mise en marche et ainsi obtenir de meilleures performances de fonctionnement. Plusieurs apports peuvent être envisagés pour élargir le champ d'utilisation du banc d'essai. On pourrait entre autres, utiliser également une autre vanne pour contrôler le débit de sortie et ainsi mesurer avec précision le débit sortant. On pourrait utiliser une sous-routine d'interface multimédia pour l'animation en temps réel. On pourrait notamment développer un compensateur multivariable en utilisant la structure du PID Dual Loop et implanter un algorithme de réglage adaptatif des gains, concevoir et implanter des compensateurs neuronaux plus performants ainsi qu'utiliser la logique floue car ces techniques font partie présentement des outils de l'automatisation. Et comme application, on pourrait utiliser le module de réglage automatique du compensateur PID Dual Loop pour d'autres systèmes et notamment sur un site industriel.

RÉFÉRENCES BIBLIOGRAPHIQUES

AL-ASSADI, S. et AL-CHALABI, L. (1987). "Optimal Gain for PID Feedback". IEEE, Control Systems Magazine, Vol. 2, No.8, 16-19.

AL-SAGGAT, U. (1991). "An Approximation Technique to Tune PID Controllers". Computer Electr. Eng., Vol. 17, No. 4, 313-320.

ASTRÖM, K.J. et HÄGGLUND, T. (1984). "Automatic Tuning of Simple Regulators with Specifications on Phase and Amplitude Margins". Automatica, Vol. 20, No. 5, 645-651.

ASTRÖM, K.J. et HÄGGLUND, T. (1988).). "Automatic Tuning of PID Controllers". ISA, ISBN: 1-55617-081-5.

ASTRÖM, K.J. et HÄGGLUND, T. (1992). "Tuning of PID Controllers". ISA, ISBN: 1-55617-081-5.

ASTRÖM, K.J., HÄGGLUND, T., HANG, C.C. et HO, W.K. (1993). "Automatic Tuning Adaptation for PID Controllers". Control Eng. Practice, Vol. 1, No. 4, 699-714.

ASTRÖM, K.J., HÄGGLUND, T. et WALLENBORG, A. (1993). "Automatic Tuning of Digital Controllers with Applications to HVAC Plants". Automatica, Vol. 29, No. 5, 1333-1343.

ASTRÖM, K.J., HANG, C.C. et LIM, B.C. (1994). "A New Smith Predictor for Controlling a Process with an Integrator and Long Dead-Time". IEEE Transactions on Automatic Control, Vol. 29, No. 2.

ASTRÖM, K.J., HANG, C.C., PERSSON, P. et HO, W.K. (1992). "Towards Intelligent PID Control". Automatica, Vol. 28, No. 1, 1-9.

ASTRÖM, K.J. et WITTENMARK, B. (1984). "Computer Controlled Systems". Prentice-Hall.

BALCHEN, J.G., LIE, B. et SOLBERG, I. (1988). "Internal Decoupling in Non-linear Process Control". Model. Identif. Control, Vol. 9, 137-148.

BARTON, G., CHAN, W. et PERKINS, J. (1991). "Interaction Between Process Design and Process Control". Journal Proc. Control 1, 161-170.

BORNE, P. et DAUPHIN, G. (1993). "Analyse et régulation des processus industriels". Éditions Technip, Tome 1, ISBN: 7108-0639-8.

BORTOLOTTI, G., DESAGES, A. et ROMAGNOLI, J.A. (1989). "Automatic Tuning of PID Controllers Through Response Optimization over a Finite Time Horizon". Chem. Eng. Comm., Vol. 86, 17-29.

BRISTOL, E. (1966). "On a New Measure of Interaction for Multivariable Process Control", IEEE, Trans. Auto. Control, Vol. AC-11, 133-134.

BROGAN, W. (1991). "Modern Control Theory". Prentice-Hall.

CHEUNG, L.S. (1988). "A New Automated Optimal Tuning Strategy for a PID Controller". Allied Fiber Plastics.

CHO, D., KATO, Y et SPILMAN, D (1993). "Sliding Mode and Classical Controllers". IEEE Control Systems, Vol. 3, No.7, 42-48.

COHEN, G.H. et COON, G.A. (1953). "Theoretical Consideration of Retarded Control". Trans. ASME, Vol. 75, 827-834.

CORNIELES, E. et BOUGERET, C. (1997). "Comparaison expérimentale de différentes techniques de réglage du régulateur PID et PID Dual Loop". École Polytechnique, Département de génie électrique et de génie informatique, Rapport EPM/RT-97/19.

CORNIELES, E. et BOUGERET, C. (1997). "Développement d'un module de réglage automatique du régulateur PID Dual Loop". École Polytechnique de Montréal, Département de génie électrique et de génie informatique, Rapport EPM/RT-97/24.

CORNIELES, E. et CONEIM, E. (1995). "Asservissement de niveau d'un réservoir d'eau avec le contrôleur PID Dual Loop". École Polytechnique de Montréal, Département de génie électrique et de génie informatique, Rapport EPM/RT-95/21.

D'AZZO, J.J. ET HOUPIS, C.H. (1981). "Linear Control Systems". McGraw Hill.

DE SANTIS, R.M. (1989). "An Adaptive PI/Sliding-Mode Controller for a Speed Drive". ASME Journal of Dynamic Systems Measurement and Control, Vol. 11, 409-415.

DE SANTIS, R.M. (1994). "A Novel PID Configuration for Speed and Position control". ASME Journal of Dynamic Systems Measurement and Control, Vol. 116, 542-549.

DE SANTIS, R.M. et CONAN, J. (1977). "Practical Sensitivity Reduction Tests with Application to Power Systems". International Journal of System Science, Vol. 8, No. 9, 1067-1080.

DE SANTIS, R.M. et PORTER, W.A. (1974). "Circle Type Conditions for Sensitivity Reduction". Automatic Control Theory and Practice, Vol. 2, No. 2, 26-36.

FRANKLIN, G., POWEL, J. et NAEIMI, A. (1991). "Feedback Control of Dynamic Systems". Addison-Wesley.

FRUEHAUF, P.S., CHIEN L. et LAURITSEN, D. (1994). "Simplified IMC-PID Tuning Rules". ISA Transaction, Vol. 33, 43-59.

GARCIA, C. et MORARI, M. (1982). "Internal Model Control". Ind. Eng. Chem. Process. Des.Dev., Vol. 21, 308.

GAWTHROP, P.J. (1986). "Self-Tuning PID Controllers: Algorithms and Implementations". IEEE Trans. On Automatic Control, Vol. AC-31, No. 3, 201-208.

GAWTHROP, P.J. et NOMIKOS, P. (1990). "Automatic Tuning of Commercial PID Controllers for Single-Loop and Multi-Loop Applications". IEEE Control Systems

Magazine, Vol. 34, No.100, 34-42.

GERRY, J.P. (1987). "A Comparison of PID Control Algorithms". Control Engineering, Mars 1987, 102-105.

GILLIAM, K. et VINANTE, C (1992). "Tuning PID Controllers for Multivariable Processes". Rev. Téc. Ing. Universidad del Zulia, Venezuela, Vol. 15, No. 3, 181-187.

HÄGGBLOM, K.E. (1993). "Experimental Comparison of Conventional and Nonlinear Model-Based Control of a Mixing Tank". American Chemical Society, Ind. Eng. Chem. Res., Vol. 32, 2653-2661.

HÄGGLUND, T. et ASTRÖM, K.J. (1991). "Industrial Adaptive Controllers Based on Frequency Response Techniques". Automatica, Vol. 27, No. 4, 599-609.

HANG, C.C., ASTRÖM, K.J. et HO, W.K. (1991). "Refinements of the Ziegler-Nichols Tuning Formula". IEEE Proceeding, Vol. 138, No. 2, 111-118.

HANG, C.C., ASTRÖM, K.J. et HO, W.K. (1993). "Relay Auto-Tuning in the Presence of Static Load Disturbance". Automatica, Vol. 29, No. 2, 563-564.

HANG, C.C., HO, W. et CAO, L.S. (1994). "A Comparison of Two Design Methods

for PID Controllers". ISA Transaction, Vol. 33, 147-151.

HANG, C.C. et SIN, K.K. (1991). "A Comparative Performance Study of PID Auto-Tuners". IEEE Control Systems, Vol. 4, No.91, 41-51.

HENSON, M.A. et SEBORG, D.E. (1990). "Input-Output Linearization of General Non linear Processes". AICHE J., Vol. 36, 1753-1757.

KRAVARIS, C. et CHUNG, C.B. (1987). "Non Linear State Feedback Synthesis by Global Input/Output Linearization". AICHE J., Vol. 33, 592-603.

KRAVARIS, C. et SOROUSH, M. (1990). "Synthesis of Multivariable Non linear Controllers by Input/Output Linearization". AICHE J., Vol. 36, 249-264.

KUO, B.C. (1995). "Automatic Control Systems". Prentice-Hall, Seventh Edition.

LEVA, A., MAFFEZZONI, C. et SCATTOLINI, R. (1994). "Self-Tuning PI-PID Regulators for Stable Systems with Varying Delay". Automatica, Vol. 30, No. 7, 1171-1183.

LUYBEN, W.L. et VINANTE, C. (1972). "Experimental Studies of Distillation Decoupling". Ken Teollisuus, Vol. 29, 499.

MARTIN, T.E. (1995). "Process Control". McGraw-Hill.

MCMILLAN, G.K. (1990). "Tuning and Control Loop Performance: A Practitioner's Guide", ISA.

MUSH, H.E. et STEINER, M. (1995). "Robust PID Control for an Industrial Distillation Column". IEEE Control System, Vol.2, No,7, 46-55.

NAKAMOTO, K. et WATANABE, N. (1991). "Multivariable Control Experiments of Non-Linear Chemical Process using Non-Linear Feedback Transformation". Journal Process Control, Vol. 1, 140-145.

OGATA, K. (1990). "Modern Engineering". Prentice-Hall.

OGUNNAIKE, B. (1986). "Controller Design for Nonlinear Process Systems via Variable Transformations". Ind. Eng. Chem. Process, Vol. 25, 241-248.

OGUNNAIKE, B. et HARMON, W. (1994). "Process Dynamics, Modeling and Control". Oxford University, Press.

OMATU, S., KHALID, M. et YUSOF, R. (1996). "Neuro-Control and its Applications". Springer, London.

PORTER, B., JONES, A.H. et McKEON, C.B. (1987). "Real-Time Expert Controllers for Plants with Actuator Non Linearities". Second IEEE International Symposium on Intelligent Control, Philadelphia.

PSALTIS, D., ATHANASIOS, S. et YAMAMURA, A. (1988). "A Multilayered Neural Network Controller". IEEE Control Systems Magazine, Vol. 8, 17-21.

RIJNSDORP, J. (1965). "Interaction for Two Variables Control Systems in Distillation Columns. I". Automatica, Vol. 1, 15-29.

RIJNSDORP, J. (1965). "Interaction for Two Variables Control Systems in Distillation Columns. II". Automatica, Vol. 1, 29-51.

RIVERA, D., MORARI, M. et SKOGESTED, S. (1986). "Internal Model Control 4 (IMC). PID Controller Design". American Chemical Society, Ind. Eng. Process Des. Dev., Vol. 25, 252-265.

RUANO, A., FLEMING, P. et JONES, D. (1992). "Connectionist Approach to PID Autotuning". IEEE, Control Systems, Proceeding-D, Vol. 39, No. 3.

RUIZ, A. et CORNIELES, E. (1997). "Asservissement en niveau et température d'un réservoir d'eau avec un contrôleur neuronal". École Polytechnique de Montréal, Département de génie électrique et de génie informatique, Section automatique, Rapport EPM/RT-98/01.

SEBORG, D., EDGAR, T. et MELLICHAMP, D. (1989). "Process Dynamics and Control". Wiley Series in Chemical Engineering.

SERFASS, C. (1987). "Contrôleur à structure variable pour une articulation robotique élastique". École Polytechnique de Montréal, Rapport technique EPM/RT-87/19.

SHINSKEY, F.G. (1988). "Process Control System". McGraw Hill, 3rd ed.

SLOTINE, J.J. (1984). "Sliding Controller Design for Non-Linear Systems". Int. J. Control, Vol. 40, No. 2.

SLOTINE, J.J. et LI, W. (1991). "Applied Non linear Control". Practice-Hall.

SMITH, O. (1957). "Closer Control of Loops with Dead Time". Chem. Eng. Prog., Vol. 53, No. 5, 217-219.

SMITH, C. et CARRIPPIO, A. (1997). "Principles and Practice of Automatic Process Control". John Wiley & Sons, Inc., Second Edition.

THOMPSON, S. (1982). "Multivariable PID Controller for Unidentified Plant". Transactions of the ASME, Vol. 104, 270-274.

VEGA, P., PRADA, C. et ALEIXANDRE, V. (1991). "Self-Tuning Predictive PID Controller". IEEE Proceedings-D, Vol. 138, No. 3. Page 1 (2 endroits).

VODA, A.A. et LANDAU, J.D. (1995). "A Method for the Auto-Calibration of PID Controller". Automatica, Vol. 31, No. 1, 41-53.

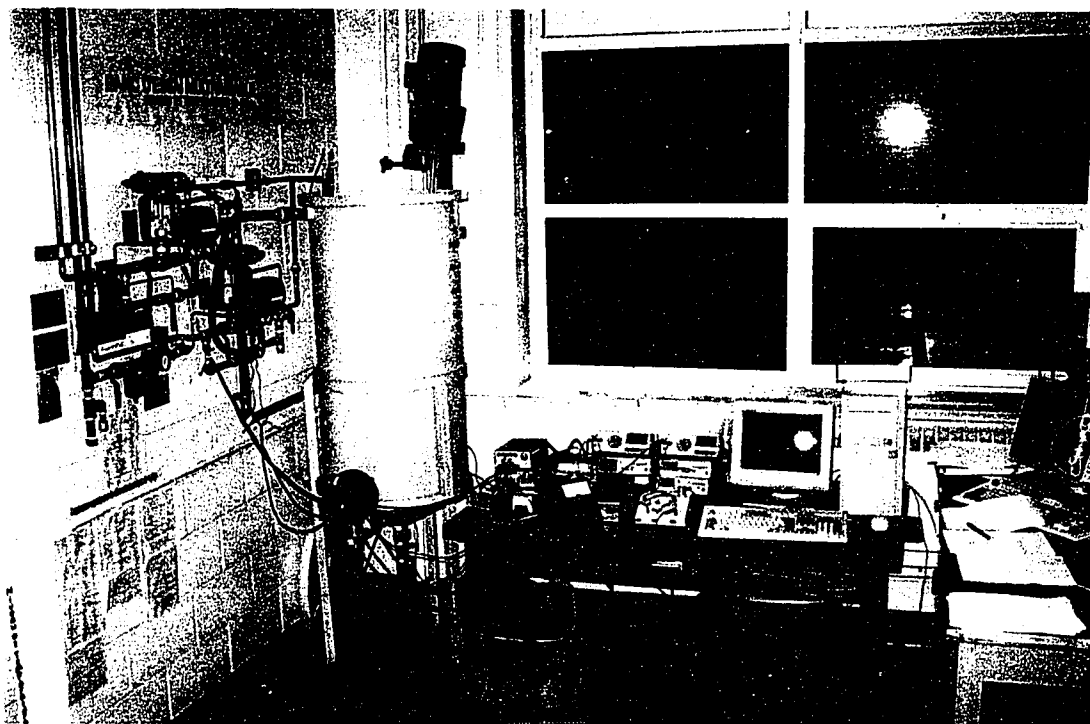
WERBOS, P.J. (1990). "Backpropagation Through Time: What it does and how to do it". Proc. IEEE, Vol. 10, 1550-1560.

WIDROW, B. et NGUYEN, D.H. (1990). "Neural Network for Self-Learning Control". IEEE Control System Magazine, Vol. 10, 18-23.

YOUJI, I., HIDEAKI, S. et HIDEKATSU, T. (1991). "A Non Linear Regulator Design in the Presence of Systems Uncertainties using Multilayered Neural Networks". IEEE Transactions on Neural Networks, Vol. 2, 410-417

ZHUANG, M. et ATHERTON, D.P. (1993). "Automatic Tuning of Optimum PID Controllers". IEEE Proceeding-D, Vol. 140, No. 3.

ZIEGLER, J.G. et NICHOLS, N.B. (1942). "Optimum Setting for Automatic Controllers". Trans. ASME, Vol. 3, 759-768.

ANNEXE 1**PHOTO DU BANC D'ESSAI POUR L'ASSERVISSEMENT
DU NIVEAU ET DE LA TEMPÉRATURE**

ANNEXE 2**LISTE DES COMPOSANTES DU BANC D'ESSAI****Liste des composantes physiques utilisées:**

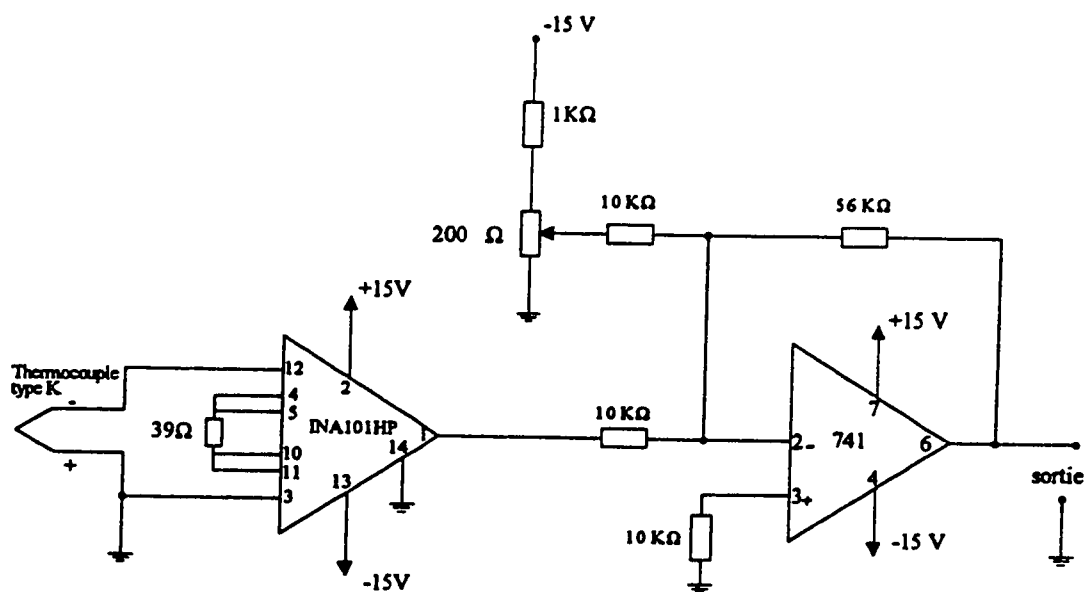
- Carte A/N, N/a Lab Master[®] Techmar
- 1 ordinateur IBM-PC, 80486, DX 33 Mhz
- Source de courant continu, KEPCO, modèle Bop 35-5(M)
- 2 convertisseurs tension/courant, Acromag[®] CSA-160T
- 3 multimètres Fluke, modèle 8010A
- 1 multimètre Fluke, modèle 8050A
- 1 multimètre Keithley, modèle 169
- 1 multimètre Fluke, modèle 8000A
- 2 vannes Foxboro[®], E691-B12-S
- Capteur de pression Foxboro[®], 847DF-MIINKE
- Agitateur Hayward Gordon, PD-14
- 1 interrupteur
- 3 capteurs de température (thermocouples) du fabricant Omega, type K, modèle SCAIN 040U-6
- 2 ampèremètres Fluke, modèle 37

Gain des capteurs:

- Kc	= 39.7 V/m	(Capteur de niveau)
- Kv1	= 0.24 l/(s.V)	(Vanne d'eau chaude)
- Kv2	= 0126 l/(s.V)	(Vanne d'eau froide)
- pv2	= 0.25 * T - 3	(Lecture de la température en V)
- R	= 6.8 K Ω	(Potentiomètre)
- R1 à R12	= 100 K Ω	(Résistance de différentes interfaces)
- Tc	= 0.25 V/°C	(Capteur de température)

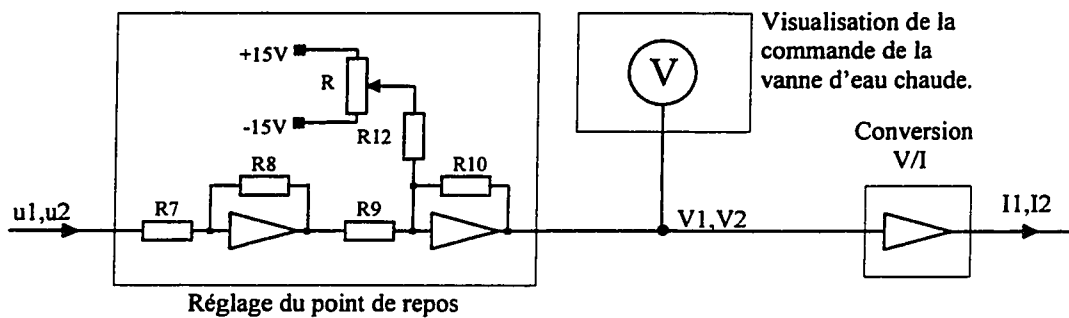
ANNEXE 3

SCHEMA COMPLET DE L'INTERFACE (4)

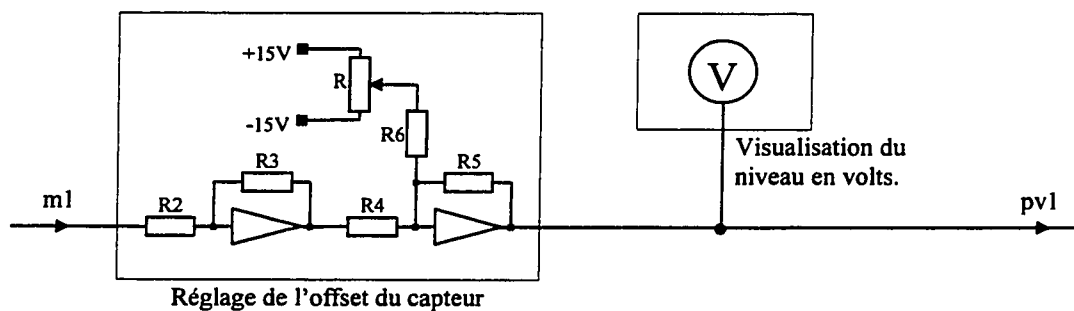


ANNEXE 4

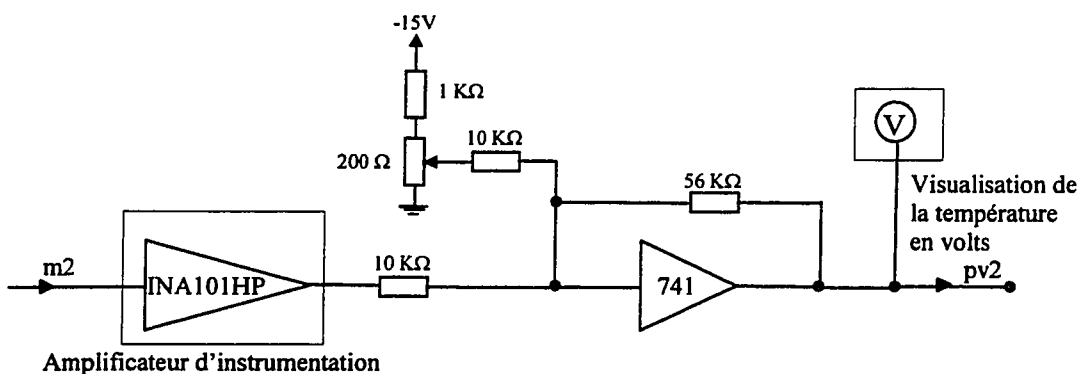
DIFFÉRENTES INTERFACES DU SYSTÈME COMPLET



Interface (1,2) entre la commande $u1, u2$ livrée par la carte Labmaster® et le courant $I1, I2$ contrôlant le débit d'eau chaude et d'eau froide



Interface (3) entre la tension $m1$ livrée par le capteur de niveau et la tension $pv1$ lue par la carte Labmaster®

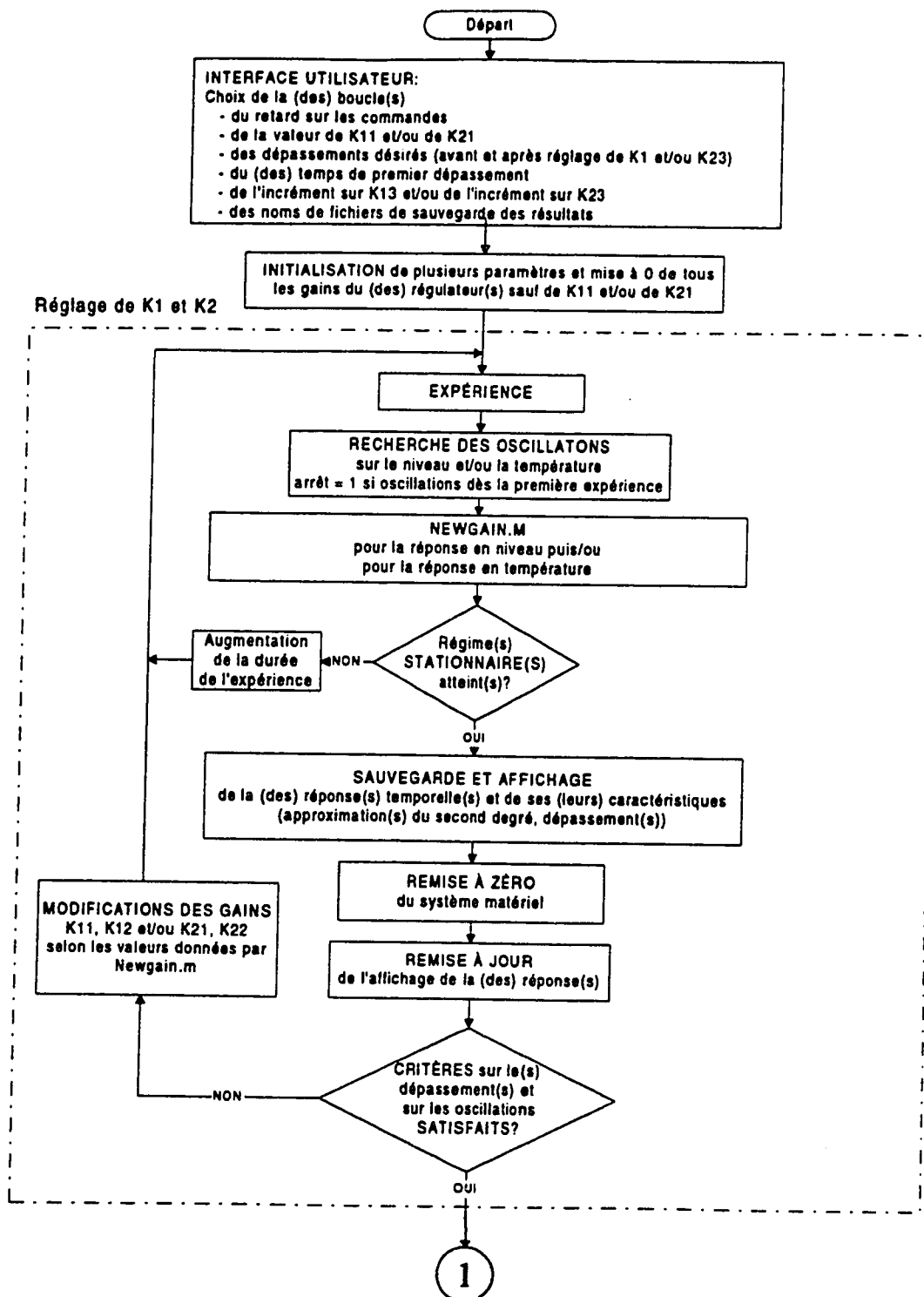


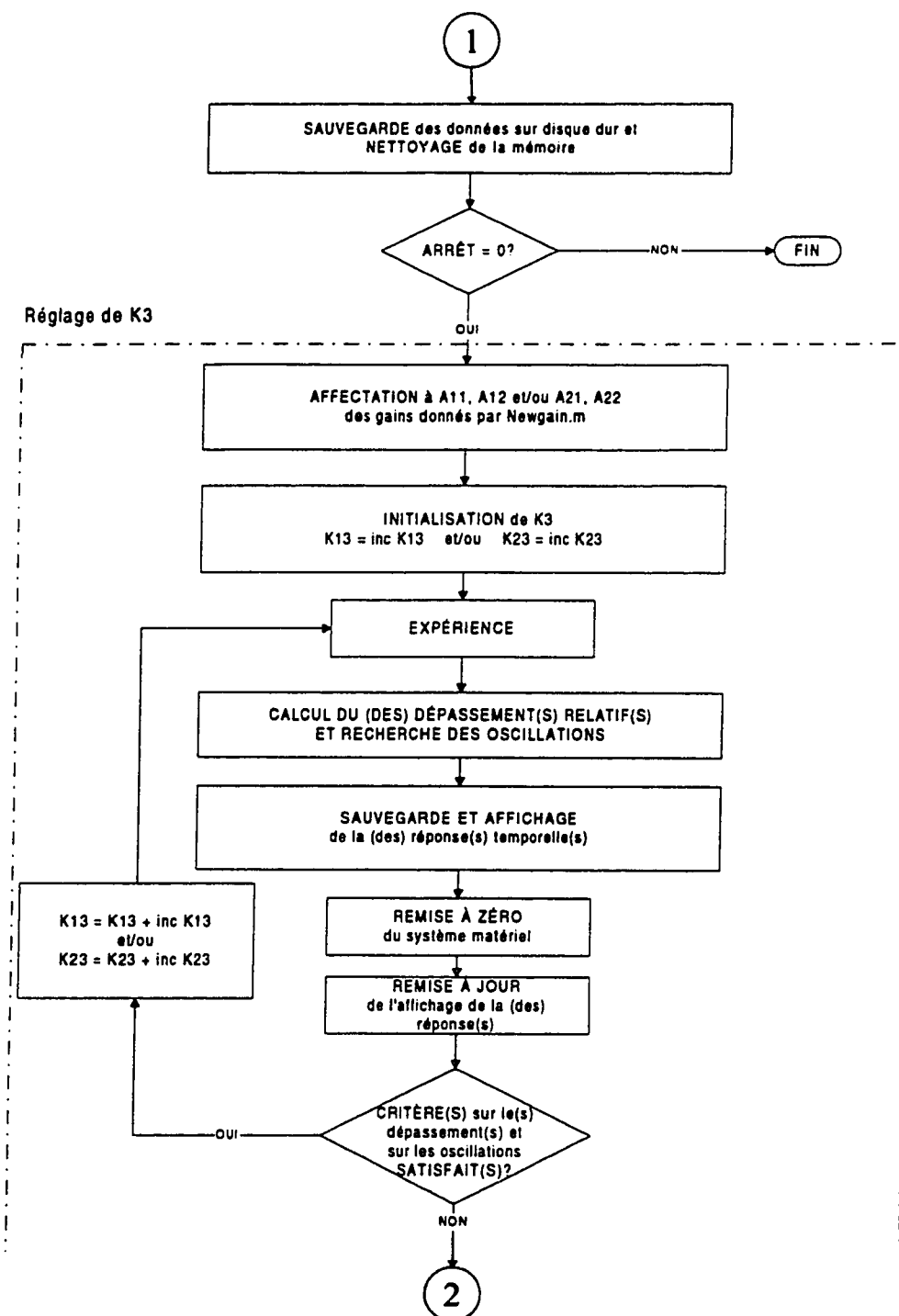
Interface (4) entre la tension $m2$ livrée par le capteur de température et la tension $pv2$ lue par la carte Labmaster® (voir à l'annexe 3 le schéma complet)

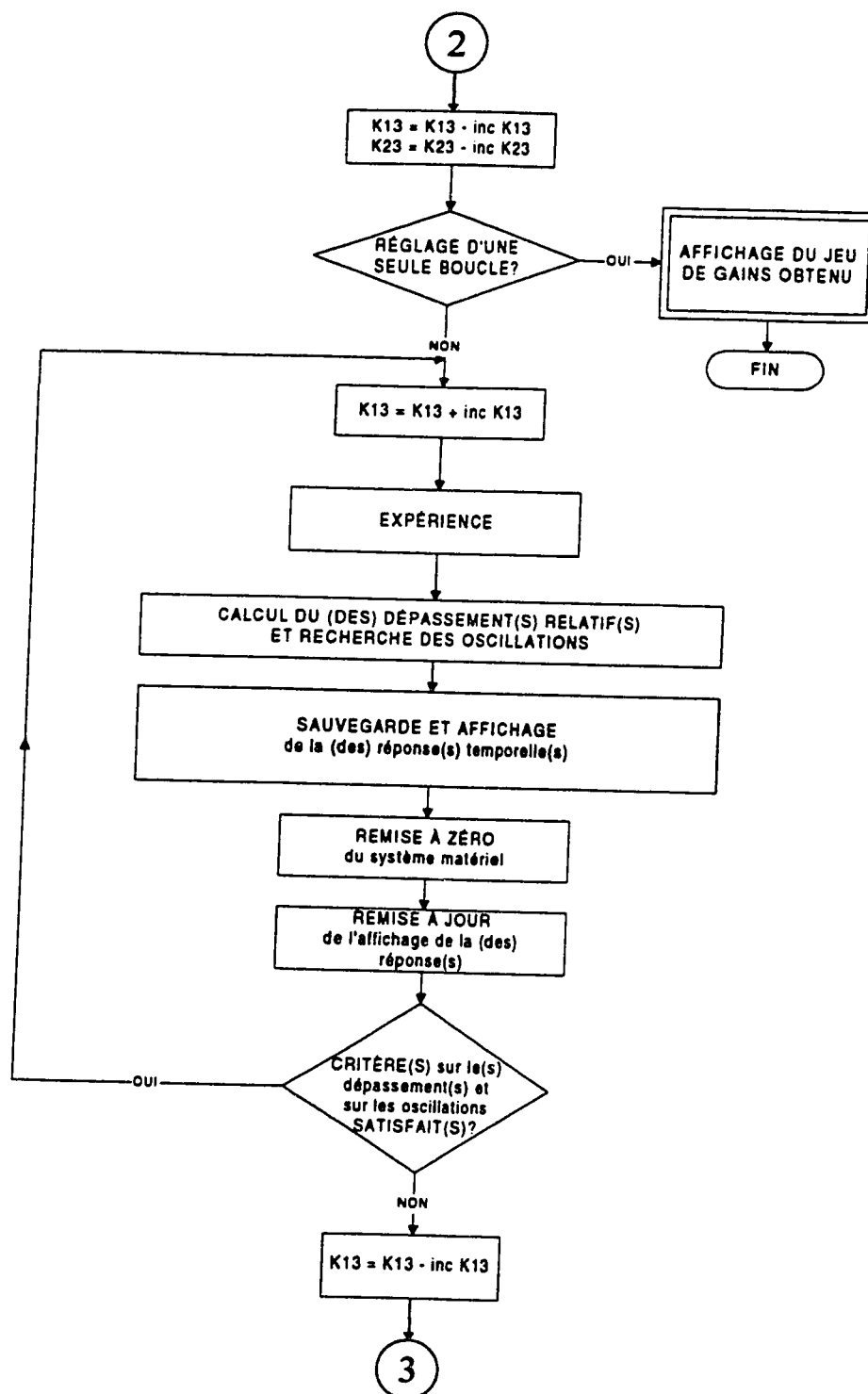
ANNEXE 5

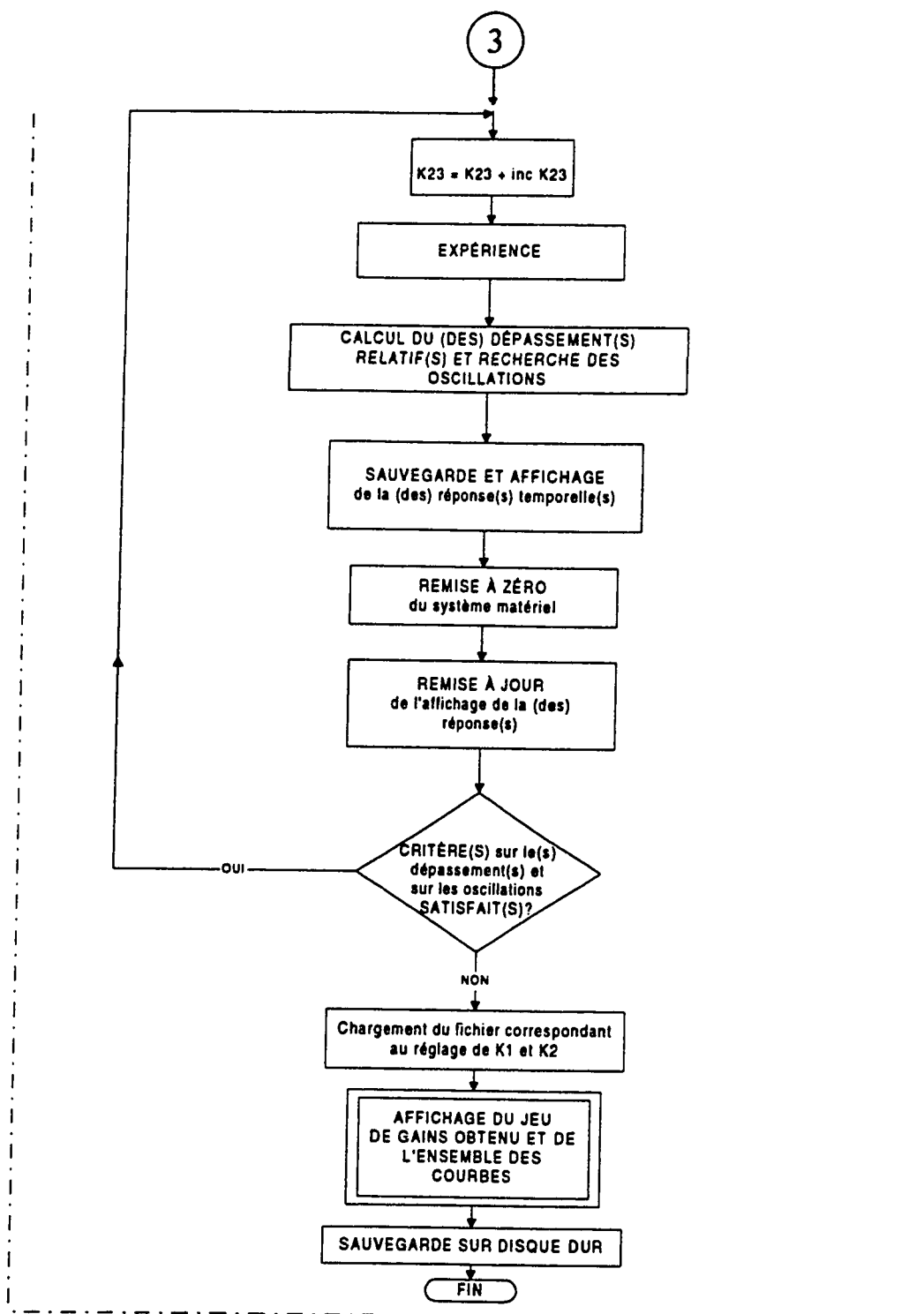
ORGANIGRAMME DU MODULE DE RÉGLAGE AUTOMATIQUE

DU COMPENSATEUR PID DUAL LOOP



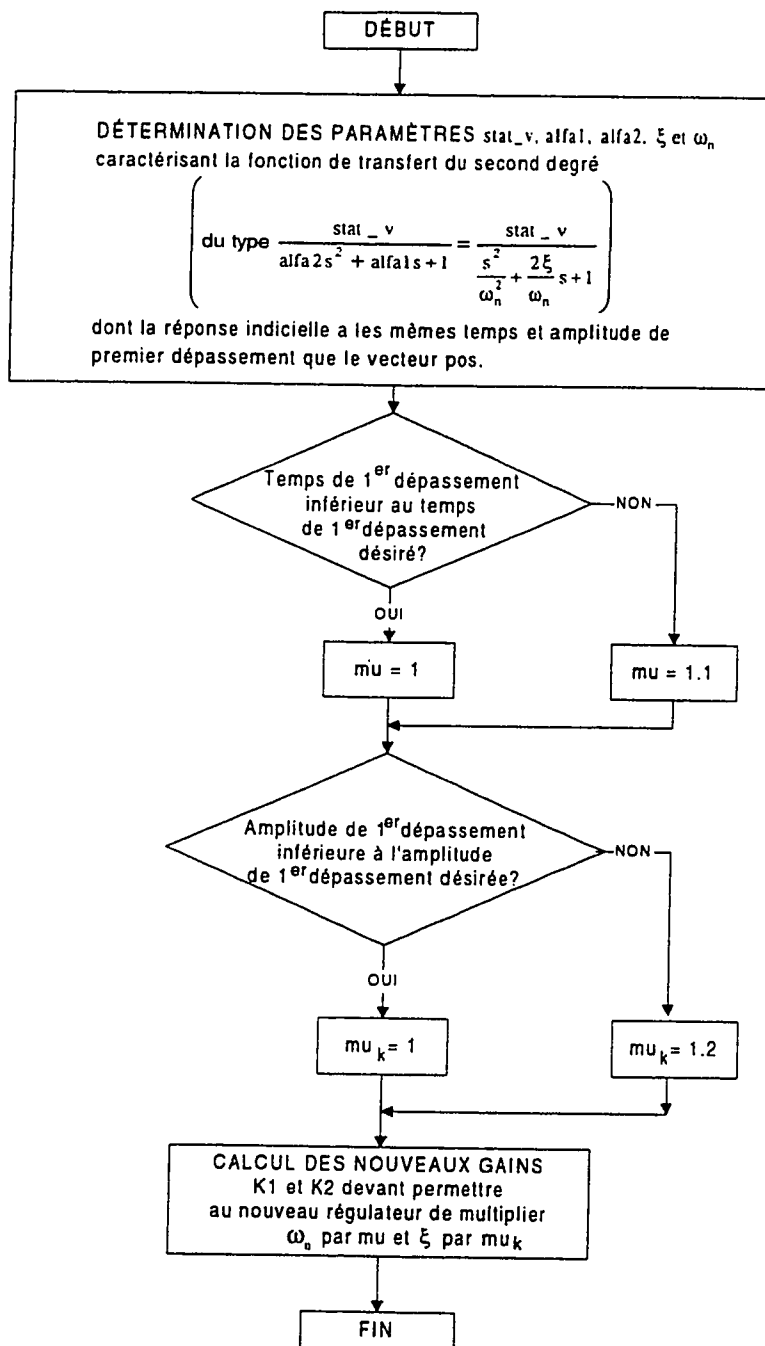






ANNEXE 6

ORGANIGRAMME DU PROGRAMME NEWGAIN.M



ANNEXE 7
TECHNIQUE DE RÉGLAGE DU PID CLASSIQUE

Techniques de réglage du PID classique

Les procédés industriels sont d'une variété extrême. Une très large classe a néanmoins pour caractéristique commune des réponses apériodiques (non oscillatoires). C'est le cas de beaucoup de procédés thermiques, chimiques, de puissance, etc. Un petit nombre de paramètres suffit généralement à caractériser leur réponse. Ainsi, les trois paramètres α , τ et R représentés sur la figure 1 constituent, en quelque sorte, la signature d'un procédé (Ogunnaike, 1994).

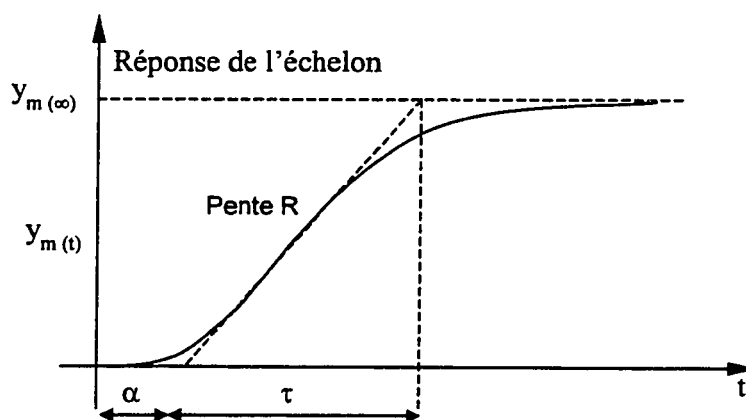


Figure 1: Réponse d'un procédé industriel

Pour calculer un compensateur, on peut se baser sur un modèle aussi représentatif que possible de la réponse observée. Un modèle simple est donné par la constante de temps retardée:

$$G(s) = \frac{K e^{-s\alpha}}{\tau s + 1}$$

On peut alors se baser sur l'approche fréquentielle (Aström et Hägglund, 1984). Plus d'un demi siècle d'expériences a enseigné aux automaticiens qu'un simple PID (ou même PI) suffisait à obtenir d'excellentes performances, surtout lorsque le rapport τ/α est grand (supérieur à 5 ou 10). Notons que les incertitudes sur les réponses (ou les paramètres α , τ et R qui correspond à la valeur de K) peuvent facilement atteindre 10% parfois 30% de plus (Hang, Aström et Ho, 1991).

Par ailleurs, le PID présente des atouts majeurs:

- il est standardisé du point de vue matériel, et on le retrouve dans toutes les technologies;
- il est standardisé sur le plan conceptuel: tous les automaticiens expérimentés connaissent l'effet spécifique de trois actions;
- le PID peut être réglé sur le site, empiriquement, par un technicien;
- en dépit de sa grande utilisation industrielle, le réglage des gains d'un compensateur PID représente encore un coût élevé.

Il existe des techniques classiques de réglage des gains du compensateur PID en utilisant les paramètres α , τ et R (Ogunnaike, 1994). On présente dans le tableau 1 les

différentes techniques de réglage du compensateur PI. On constate au tableau 1 que les expressions des paramètres sont d'un compensateur proportionnel intégral car pour

notre procédé en étude, il suffit d'un compensateur PI, où $\frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{\tau_i s} \right)$.

Tableau 1: Paramètres du compensateur proportionnel intégral pour différentes techniques de réglage (Ogunnaike, 1994).

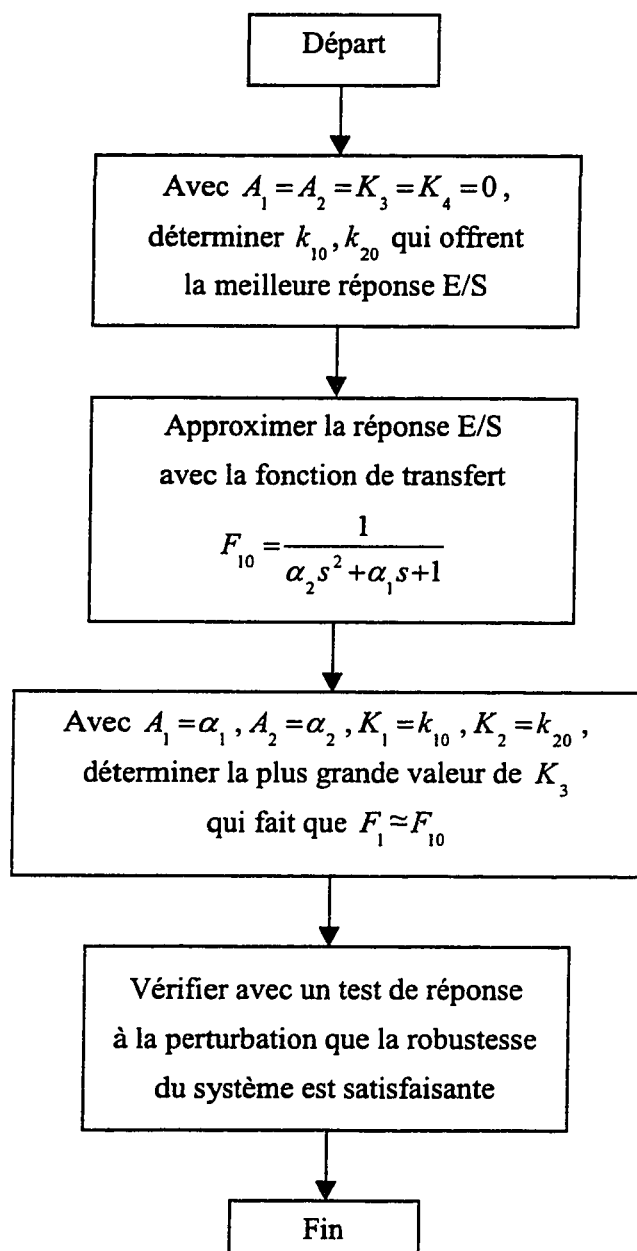
Paramètres du compensateur proportionnel intégral		
Type de réglage	K_p	τ_i
Ziegler-Nichols	$\frac{0.9}{K_r} \left(\frac{\tau}{\alpha} \right)$	3.33α
Cohen-Coon	$\frac{1}{K_r} \left(\frac{\tau}{\alpha} \right) \left[0.9 + \frac{1}{12} \left(\frac{\alpha}{\tau} \right) \right]$	$\alpha \left[\frac{30 + 3 \left(\frac{\alpha}{\tau} \right)}{9 + 20 \left(\frac{\alpha}{\tau} \right)} \right]$
Rivera-Morari (IMC)	$\frac{\tau}{\lambda K_r}$	avec $\frac{\lambda}{\alpha} > 1.7$
ITAE (Consigne)	$\frac{0.586}{K_r} \left(\frac{\tau}{\alpha} \right)^{0.916}$	$\frac{\tau}{[1.03 - 0.165 \left(\frac{\alpha}{\tau} \right)]}$
ITAE (Perturbation)	$\frac{0.859}{K_r} \left(\frac{\tau}{\alpha} \right)^{0.977}$	$\frac{\tau}{0.674} \left(\frac{\alpha}{\tau} \right)^{0.680}$

Les valeurs K et K_r pour les paramètres du compensateur proportionnel intégral se définissent ainsi:

- K : gain de la fonction de transfert correspondant à la pente de la réponse en boucle ouverte.
- K_r : valeur en régime permanent de la réponse en boucle ouverte.

ANNEXE 8

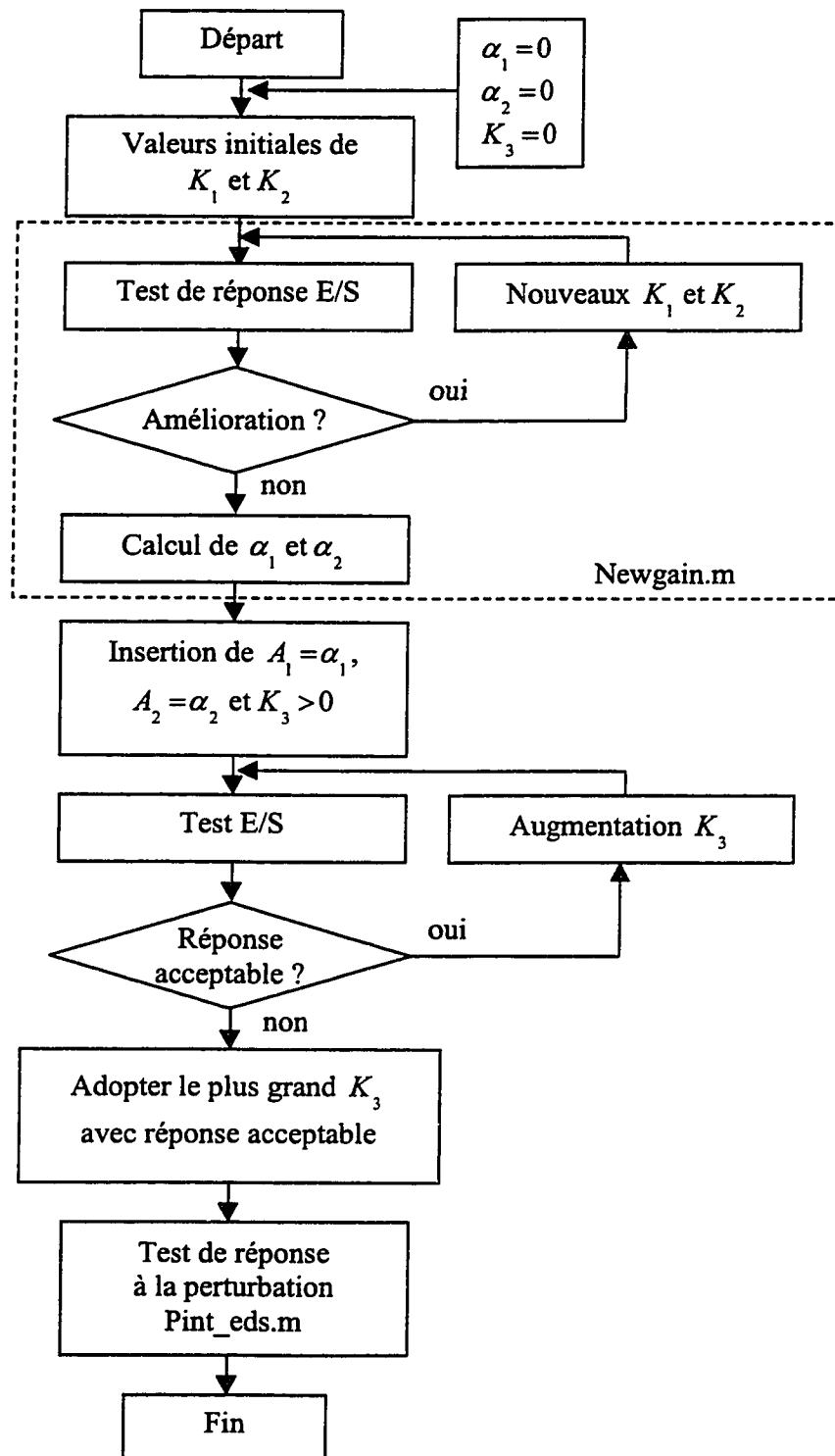
**ORGANIGRAMME DE LA SOLUTION DU PROBLÈME
EN UTILISANT LE PID DUAL LOOP**



ANNEXE 9

PROCÉDURE DE RÉGLAGE DES PARAMÈTRES

DU COMPENSATEUR PID DUAL LOOP



ANNEXE 10
PROGRAMMES DU MODULE DE RÉGLAGE AUTOMATIQUE
DU COMPENSATEUR PID DUAL LOOP

```

%*****
% programme Pintpri2.m
% Ce programme écrit avec Matlab 4.2c permet de déterminer
% automatiquement(auto-tuning) les gains du (des) régulateurs
% Dual Loop du réservoir en fonction des critères sur la répo
% nse désirés par l'utilisateur(temps et amplitude du premier
% dépassement). Ce programme a été utilisé avec un pentium166
% muni de 32Mo RAM et une carte d'entrées/sorties Labmaster
% fait appel aux sous-programmes:
% -pintauto.m (stratégie de contrôle),
% -newgain.m (identification de la réponse)
% -raz.m (remise à zéro)
% -rech_osc.m (recherche d'oscillations)
% -reglak3.m (réglage du gain k3 du PID Dual Loop)
%*****

fprintf('*****
\n');
fprintf('    Réglage automatique des gains du PID Dual Loop.
\n');
fprintf('    Asservissement de niveau et température.
\n');
fprintf('    Version: 12/06/97. par: EC/CB/RMDS
\n');
fprintf('*****
\n');

clear

%=====
%
%                INTERFACE UTILISATEUR
%=====

fprintf('Réglage des gains de la boucle de: \n');
fprintf('1.-Niveau \n');
fprintf('2.-Température \n');
fprintf('3.-Niveau et Température \n');
F=input('');

fprintf('\n Retard sur la commande? \n');
fprintf('0.-non \n');
fprintf('1.-oui \n');
ret=input('');

if ret==1,
    fprintf('\n Valeur du retard (en s) ? \n');
    retard=2*input('');

```

```

end;

if ((F==1)|(F==3)),
    fprintf('\n ***** \n'
);
    fprintf(' NIVEAU:\n');
    fprintf('\nValeur initiale de K11?\n');
    K11=input('');
    fprintf('\n Instant de 1er dépassement (s)? \n');
    instn=input('');
    fprintf('\n Amplitude relative du 1er dépassement:\n'
);
    fprintf('-après le réglage de k1 et k2 ?:\n');
    ampln12=input('');
    fprintf('\n-après le réglage de k3?:\n');
    ampln3=input('');
    fprintf('\n Incrément sur K13? \n');
    incK13=input('');
end;

if ((F==2)|(F==3)),
    fprintf('\n ***** \n'
);
    fprintf(' TEMPÉRATURE:\n');
    fprintf(' \n Instant de 1er dépassement (s)? \n');
    instt=input('');
    fprintf('\nValeur initiale de K21?\n');
    K21=input('');
    fprintf(' \n Amplitude relative du 1er dépassement:\n'
);
    fprintf('-après le réglage de k1 et k2 ?:\n');
    amplt12=input('');
    fprintf('\n-après le réglage de k3?:\n');
    amplt3=input('');
    fprintf('\n Incrément sur K23? \n');
    incK23=input('');
end;

fprintf('\nnom des fichiers de sauvegarde:\n');
fprintf('(exemple: 'c:\matlab\reserve\toto')\n');
fprintf('\n-pour la première phase du réglage?\n');
nom0=input('');
fprintf('\n-pour la seconde phase du réglage?\n');
nom=input('');
fprintf('\nQuand vous êtes prêt\n');
fprintf('et que le système est stationnaire, tapez return\n');
;
keyboard

```

```

fprintf('=====\\n');

%=====
%                               INITIALISATION
%=====

z=1;           % numéro de l'expérience
retour=0;      % utilisé dans le sous programme RAZ.m
w1=0;          % offsets sur les commandes des vannes
w2=0;

if F==1,
    K12 = 0;
    K13 = 0;
    K14 = 0;
    A11 = 0;
    A12 = 0;
    K21 = 0;
    K22 = 0;
    K23 = 0;
    K24 = 0;
    A21 = 0;
    A22 = 0;

    tfpan=1000;
    instt=1;
    tfpat=0;
    amplt12=1000;
    amplt3=1000;
end

if F==2,
    K11 = 0;
    K12 = 0;
    K13 = 0;
    K14 = 0;
    A11 = 0;
    A12 = 0;
    K22 = 0;
    K23 = 0;
    K24 = 0;
    A21 = 0;
    A22 = 0;

    tfpat=1000;
    instn=1;
    tfpan=0;
    ampln12=1000;

```



```

        ampln3=1000;
end

if F==3,
    K12 = 0;
    K13 = 0;

    K14 = 0;
    A11 = 0;
    A12 = 0;
    K22 = 0;
    K23 = 0;
    K24 = 0;
    A21 = 0;
    A22 = 0;

    tfpan=1000;
    tfpat=1000;

end;

%=====
%               réglage de k1, k2, alpha1 et alpha2
%=====

overshootn=100;
overshoott=100;
osciln=0;
oscilt=0;

while (((tfpan>instn)|(tfpat>instt)|(overshootn>ampln12)|(overshoott>amplt12)) & (osciln==0) & (oscilt==0)),

    nb_pts=1500;
    if F==3
        nb_pts=2000;
    end
    ref1=2;
    ref2=5;

    fprintf('\nEXPÉRIENCE %g.\n',z)

    if ((F==1)|(F==3))
        gain1n(z)=K11;
        gain2n(z)=K12;
        fprintf('K11= %g.\t',K11);
        fprintf('K12= %g.\n',K12);
    end
end

```

```

        if ((F==2)|(F==3))
            gain1t(z)=K21;
            gain2t(z)=K22;
            fprintf('K21= %g.\t',K21);
            fprintf('K22= %g.\n',K22);
        end

        pintauto;
%-----
%Recherche des oscillations
%-----
        if ((F==1)|(F==3)),
            pos=pv1;
            rech_osc;
            osciln=oscil;
            if osciln==1,
                fprintf('sur le niveau\n');
            end;
        end;

        if ((F==2)|(F==3)),
            pos=pv2-pv2(1);
            rech_osc;
            oscilt=oscil;
            if oscil==1,
                fprintf('sur la température\n');
            end;
        end;

%-----

        arret=0;
        if ((osciln==1)|(oscilt==1))&(z==1),
            fprintf('Oscillations dès la lère expérience\
n');
            fprintf('donc arrêt du programme après la RAZ
\n');
            fprintf('Essayez de nouveaux gains initiaux\n
');
            arret=1;
        end

%-----

        if ((osciln==0)&(oscilt==0)),
            tech=T;
%-----

```

```

% BOUCLE DE NIVEAU
%-----

    if ((F==1)|(F==3)),

        k10n=K11;% gains utilisés pour le ré-
        k20n=K12;% glage de K3

        pos=pv1;% initialisation des paramè-
        k1=K11; % tres pour utiliser le
        k2=K12; % sous-programme newgain.m
        Fs=0;
        if F==3,
            Fs=3;
            F=1;
        end;

        newgain;

%-----
% on refait l'expérience sur une plus longue durée
% si le régime stationnaire n'a pas été atteint
%-----

        nbre=nb_pts;
        nbrel=nb_pts;
        while srnawrt==1;
            fprintf('expérience niveau tr
op courte');

            raz;
            nbre=nbre+500;
            nb_pts=nbre;
            pintauto;
            pos=pv1;
            newgain
        end
        nb_pts=nbrel;

%-----
% on redonne à F sa valeur initiale
%-----

        if Fs==3,
            F=Fs;
            Fs=0;
        end;

%-----
%Sauvegarde et affichage des données
%-----

```

```

k1n=k1;          % on sauvegarde les
k2n=k2;          % paramètres
                  % concernant le
                  % le niveau
alfa_10n=alfa_10;
alfa_20n=alfa_20;

fprintf('\nIdentification de la répon
se en niveau \n');

a1n(z)=alfa_10n;
a2n(z)=alfa_20n;
depassementn(z)=overshoot;
tpdn(z)=tfpa;

overshootn=overshoot;
tfpan=tfpa;
fprintf('A11= %g.\t',alfa_10n);
fprintf('A12= %g.\n',alfa_20n);
fprintf('depassement= %g.\n',overshoo
tn);

fprintf('Tps de 1er dépassement= %g.\
n',tfpan);

end;

%-----
% BOUCLE DE TEMPÉRATURE
%-----

if ((F==2) | (F==3)),

    k10t=K21;% gains utilisés pour le
    k20t=K22;% réglage de k3

    pos=pv2-pv2(1); % initialisation des
    k1=K21;          % paramètres pour
    k2=K22;          % le sous-programme
    Fs=0;            % newgain.m
    if F==3,
        Fs=3;
        F=2;
    end;

    newgain;

%-----
% on refait l'expérience sur une plus longue durée
% si le régime stationnaire n'a pas été atteint

```

```

%-----
                                nbre=nb_pts;
                                nbrel=nb_pts;
                                while srnawrt==1;
                                    fprintf('expérience températu
re trop courte');
                                    raz;
                                    nbre=nbre+500;
                                    nb_pts=nbre;
                                    pintauto;
                                    pos=pv2-pv2(1);
                                    newgain
                                end
                                nb_pts=nbrel;

%-----
% on redonne à F sa valeur initiale
%-----
                                if Fs==3,
                                    F=Fs;
                                    Fs=0;
                                end;

%-----
% Sauvegarde et affichage des données
%-----
                                k1t=k1; % on sauvegarde les paramètre
s
                                k2t=k2; % concernant la température
                                alfa_10t=alfa_10;
                                alfa_20t=alfa_20;

                                fprintf('\nIdentification de la répon
se en température \n');
                                alt(z)=alfa_10t;
                                a2t(z)=alfa_20t;
                                depassementt(z)=overshoot;
                                tpdt(z)=tfpa;

                                overshoott=overshoot;
                                tfpat=tfpa;

                                fprintf('A21= %g.\t',alfa_10t);
                                fprintf('A22= %g.\n',alfa_20t);
                                fprintf('dépassement= %g.\n',overshoo
tt);
                                fprintf('Tps de 1er dépassement= %g.\n',tfpat);

```

```

end;
end;

%=====
%                                AFFICHAGE DE LA OU DES RÉPONSES
%=====
    if F==1,
        subplot(224),plot(pv1); grid
        subplot(211),plot(pv11); grid
    end;

    if F==2,
        subplot(224),plot(pv2); grid
        subplot(211),plot(pv21); grid
    end;

    if F==3,
        subplot(211),plot(pv11); grid
        subplot(212),plot(pv21); grid
    end;

    pause(5);

    raz;    % Remise à zéro

    if F==1,
        subplot(211),plot(pv11); grid
    end;

    if F==2,
        subplot(211),plot(pv21); grid
    end;

    if F==3,
        subplot(211),plot(pv11); grid
        subplot(212),plot(pv21); grid
    end;

    pause(5);

%=====
%                                PASSAGE AU PAS SUIVANT
%=====
    if ((F==1)|(F==3)),
        K11=k1n;
        K12=k2n;
    end;

```

```

        if ((F==2)|(F==3)),
            K21=k1t;
            K22=k2t;
        end;

        z=z+1;
    end;

%=====
%      Stockage des données et nettoyage de la mémoire
%=====

eval(['save ' nom0 ' pv11 pv21 comm1 comm2']);

clear pv11;
clear pv21;
clear comm1;
clear comm2;
pack;
pause(5);

z0=z-1;
z=1;

%=====
%                                réglage de k3
%=====
if arret==0
    fprintf('\n=====
=====\\n');
    fprintf('RÉGLAGE DE k3');

    if ((F==1)|(F==3))
        K11=k10n;
        K12=k20n;
        A11=alfa_10n;
        A12=alfa_20n;
    end

    if ((F==2)|(F==3)),
        K21=k10t;
        K22=k20t;
        A21=alfa_10t;
        A22=alfa_20t;
    end

    Fr=F;

```

```

    reglak3;

    if ((F==1)|(F==3)),
        K13=K13-incK13;
    end;

    if ((F==2)|(F==3)),
        K23=K23-incK23;
    end;

    if (F==3),
        Fr=1;
        reglak3;
        K13=K13-incK13;

        Fr=2;
        reglak3;
        K23=K23-incK23;
    end;

%=====
%                               AFFICHAGE DES RÉSULTATS
%=====

    pv111=pv11;
    pv121=pv21;
    comm11=comm1;
    comm21=comm2;

    eval(['load ' nom0]);

    pv11=[pv11 pv111];           % Concaténation des réponses
obtenues    pv21=[pv21 pv121];   % lors des 2 phases de réglag
e           comm1=[comm1 comm11];
           comm2=[comm2 comm21];

    temps1=0.5*[1:1:max(size(pv11))];
    subplot(411),plot(temps1,pv11);
    axis([1,0.5*max(size(pv11)),floor(min(pv11)),floor(max
x(pv11))+1]);
    ylabel('niveau (V)');
    grid;

    temps2=0.5*[1:1:max(size(pv21))];
    subplot(412),plot(temps2,pv21);

```



```

        axis([1,0.5*max(size(pv21)),floor(min(pv21)),floor(max
x(pv21))+1]);
        ylabel('température (V)');
        grid;

        temps3=0.5*[1:1:max(size(comm1))];
        subplot(413),plot(temps3,comm1);
        axis([1,0.5*max(size(comm1)),floor(min(comm1)),floor(
max(comm1))+1]);
        ylabel('u1 (V)');
        grid;

        temps4=0.5*[1:1:max(size(comm2))];
        subplot(414),plot(temps4,comm2);
        axis([1,0.5*max(size(comm2)),floor(min(comm2)),floor(
max(comm2))+1]);
        xlabel('temps (s)');
        ylabel('u2 (V)');
        grid;

        fprintf('\n Les gains trouvés sont:\n');
        fprintf('K11= %g.\n',K11);
        fprintf('K12= %g.\n',K12);
        fprintf('K13= %g.\n',K13);
        fprintf('A11= %g.\n',A11);
        fprintf('A12= %g.\n',A12);
        fprintf('K21= %g.\n',K21);
        fprintf('K22= %g.\n',K22);
        fprintf('K23= %g.\n',K23);
        fprintf('A21= %g.\n',A21);
        fprintf('A22= %g.\n',A22);

%=====
%
%                               SAUVEGARDE DES RÉSULTATS
%=====
        if arret==1
            eval(['save ' nom ' F pv11 pv21 comm1 comm2 K
11 K12 K13 A11 A12 K21 K22 K23 A21 A22 gain1n gain2n']);
        else
            if z==1
                depassement3n=0;
                depassement3t=0;
            end
            if F==1
                eval(['save ' nom ' F pv11 pv21 comm1
comm2 K11 K12 K13 A11 A12 K21 K22 K23 A21 A22 gain1n gain2n
a1n a2n tpdn depassementn depassement3n']);
            end
        end

```

```

        if F==2
            eval(['save ' nom ' F pv11 pv21 comm1
comm2 K11 K12 K13 A11 A12 K21 K22 K23 A21 A22 gain1t gain2t
alt a2t tpdtd depassementt depassement3t']);
        end
        if F==3
            eval(['save ' nom ' F pv11 pv21 comm1
comm2 K11 K12 K13 A11 A12 K21 K22 K23 A21 A22 alfa_10 alfa_2
0 gain1n gain2n aln a2n gain1t gain2t alt a2t tpdn tpdtd depas
sementn depassementt depassement3n depassement3t']);
        end
    end
end

ecr_na (0,0);    % remise à zéro des commandes
ecr_na (1,0);

```

```

%=====
% Programme Pintauto.m
% SOUS-PROGRAMME DU PROGRAMME PINTPRI2.m CONSTITUANT LE RÉGU
% LATEUR PID DUAL LOOP NUMÉRIQUE
% Ce programme permet l'acquisition et l'écriture de données
% avec une carte Labmaster
% Version : 27 mai 1997 par EC/CB/RMDS
%=====

retl=ret;
if (retour==1),
    ret=0;
    fprintf('\n REMISE À ZERO \n');
end;

T = 0.5;

alpha1 = 1.723;
alpha11=1/(1+alpha1);
alpha12=alpha1*alpha11;

alpha21=0.36;
alpha22=-0.64;

pv1 = zeros(1,nb_pts);
pv2 = zeros(1,nb_pts);
u1 = zeros(1,nb_pts);
u2 = zeros(1,nb_pts);

pvv1 =zeros(1,nb_pts);
pvv2 =zeros(1,nb_pts);

err1 = zeros(1,nb_pts);
err2 = zeros(1,nb_pts);

int_err1= zeros(1,nb_pts);
int_err2= zeros(1,nb_pts);

der1=zeros(1,nb_pts);
der2=zeros(1,nb_pts);

U1 = zeros(1,nb_pts);
U2 = zeros(1,nb_pts);

fprintf('\n SOYEZ PATIENT!\n');
est=0;
fprintf('0---1---2---3---4---5---6---7---8---9---10\n');

```

```

init_lab;          % INITIALISATION CONVERTISSEUR

dep_horl;          % DÉPART HORLOGE

pv1(1)=lire_an(7);
ref1=ref1-pv1(1);
consigne1 = ref1*ones(1,nb_pts);

pv3(1)=lire_an(6);
pv2(1)=pv3(1);
ref2=ref2-pv2(1);
consigne2 = ref2*ones(1,nb_pts);

att_horl (1000*T);    % temporisation de T seconde

%=====
%                               BOUCLE DE RÉGULATION
%=====

for i=2:nb_pts,

    dep_horl;

    if (100*i/nb_pts)>(est+2.45),    % Indication graphi-
        est=est+2.5;                % que de l'avance-
        fprintf('*');               % ment de l'expéri-
    end;                             % ence

    pv1(i) = lire_an(7);            % ACQUISITION DES DONNÉES DE
    pv3(i) = lire_an(6);            % LA CARTE

%niveau
    pvv1(i)=pv1(i)-pv1(1);
    err1(i)= consigne1(1,i)-pvv1(i);
    int_err1(i)=int_err1(i-1)+err1(i)*T;
    der1(i)=(pvv1(i)-pvv1(i-1))/T;
    U1(i)=K11*err1(i)+K13*(int_err1(i)-A11*pvv1(i)-A12*de
r1(i))+K14*ref1-K12*der1(i);

%temperature
    pv2(i)=(T*pv3(i)+20*pv2(i-1))/(T+20);

    pvv2(i)=pv2(i)-pv2(1);
    err2(i)=consigne2(1,i)-pvv2(i);
    der2(i)=(pvv2(i)-pvv2(i-1))/T;
    int_err2(i)=int_err2(i-1)+err2(i)*T;
    U2(i)=K21*err2(i)+K23*(int_err2(i)-A21*pvv2(i)-A22*de

```

```

r2(i))+K24*ref2-K22*der2(i);

%=====
    if F==1,
        if ret==1,
            if i<=retard,
                UU1(i)=0;
            elseif i>retard,
                UU1(i)=U1(i-retard);
            end
            u11(i)=alpha11*UU1(i);
            u12(i)=alpha12*UU1(i);
            u21(i)=0;
            u22(i)=0;
        end
        if ret==0,
            u11(i)=alpha11*U1(i);
            u12(i)=alpha12*U1(i);
            u21(i)=0;
            u22(i)=0;
        end
    end
%=====

    if F==2
        if ret==1,
            if i<=retard,
                UU2(i)=0;
            elseif i>retard,
                UU2(i)=U2(i-retard);
            end
            u21(i)=alpha21*UU2(i);
            u22(i)=alpha22*UU2(i);
            u12(i)=0;
            u11(i)=0;
        end
        if ret==0,
            u21(i)=alpha21*U2(i);
            u22(i)=alpha22*U2(i);
            u12(i)=0;
            u11(i)=0;
        end
    end
end

```

```

=====
    if F==3,
        if ret==1,
            if i<=retard,
                UU1(i)=0;
                UU2(i)=0;
            elseif i>retard,
                UU1(i)=U1(i-retard);
                UU2(i)=U2(i-retard);
            end
            u11(i)=alpha11*UU1(i);
            u12(i)=alpha12*UU1(i);
            u21(i)=alpha21*UU2(i);
            u22(i)=alpha22*UU2(i);
        end
        if ret==0,
            u11(i)=alpha11*U1(i);
            u12(i)=alpha12*U1(i);
            u21(i)=alpha21*U2(i);
            u22(i)=alpha22*U2(i);
        end
    end

=====
% commandes avec prédécouplage

    u1(i) = u11(i) + u21(i);
    u2(i) = u12(i) + u22(i);

    ecr_na (0, u1(i)+w1);    % ENVOI DES COMMANDES A LA
    ecr_na (1, u2(i)+w2);    % CARTE

    att_horl (1000 * T);

end;

=====
%
%                               SAUVEGARDE DES DONNÉES
%
=====

if ((z==1)&(retour~=1))
    pv1(1:max(size(pv1)))=pv1;
    pv2(1:max(size(pv2)))=pv2;
    comm1(1:max(size(u1)))=u1;
    comm2(1:max(size(u2)))=u2;
else

```

```

        pv11(max(size(pv11)+1):(max(size(pv11))+max(size(pv1)
    ))=pv1;
        pv21(max(size(pv21)+1):(max(size(pv21))+max(size(pv2)
    ))=pv2;
        comm1(max(size(comm1)+1):(max(size(comm1))+max(size(u
    1))))=u1;
        comm2(max(size(comm2)+1):(max(size(comm2))+max(size(u
    2))))=u2;
    end

    fprintf(' \n');

    ret=ret1;

```

```

%=====
% Programme Rech_osc.m
% PROGRAMME POUR DÉTECTER LA PRESENCE ÉVENTUELLE D'OSCILLA-
% TIONS DANS LE VECTEUR pos (variable du procédé)
% Version 12 juin 1997 par EC/CB/RMDS
%=====

%=====
%                               INITIALISATION
%=====

N=max(size(pos));
i=30;
maxloc=0;
oscil=0;

%=====
% RECHERCHE DU PREMIER MAXIMUM LOCAL SIGNIFICATIF
%=====

while ((maxloc==0)&(i<N-30)),
    i=i+1;
    if ((pos(i+1)<=pos(i))&(pos(i-1)<=pos(i))),
        maxloc=1;
        minimum=min(pos(i-30:i+30));
        maximum=max(pos(i-30:i+30));
        dif=maximum-minimum;

        if dif<0.1,
            maxloc=0;
        end;

        if maxloc==1,
            for m=-30:30,
                if pos(i+m)>pos(i),
                    maxloc=0;
                end
            end
        end;
    end;
    %keyboard
end;

end;
%=====
% RECHERCHE DES OSCILLATIONS ÉVENTUELLES DU MAXIMUM LOCAL
%=====

if maxloc==1,
    % recherche du minimum global

```



```

        mini=min(pos(i:N));      % entre l'abscisse du 1er max
imum      k=i;                  % local retenu et la fin de l
'expérience      while (pos(k)-mini)>0.000001,
                    k=k+1;
                end;

        moy=mean(pos(k:N));      % recherche du maximum global
entre      maxi=max(pos(k:N));   % l'abscisse du minimum globa
l retenu et la fin de l'expérience
        l=k;
        while (maxi-pos(l))>0.000001,
            l=l+1;
        end;

        ondul=(maxi-mini)/moy;
        oscil=0;
        if ((ondul>0.05)&(l-k)<100),
            oscil=1;
            fprintf('\nPrésence d''oscillations\t');
        end;
end;

```

```

%=====
% Programme Raz.m
% SOUS-PROGRAMME DU PROGRAMME PINTPRI2.m
% PERMETTANT LA REMISE A ZERO AUTOMATIQUE DU RÉSERVOIR.
% Version: 15 juin 1997 par EC/CB/RMDS
%=====

%=====
% RAZ du niveau seul si F=1
% RAZ du niveau ET de la temperature si F=2 ou F=3
%=====

clear pv1;
clear pv2;
clear u1;
clear u2;

retour=1;
Fb=0;
nb_pts=2000;
ref1=0;
ref2=2;

sauv1n=K11;
K11=1;
sauv2n=K12;
K12=0;
sauv3n=K13;
K13=0;

if F==2,
    Fb=F;
    F=3;
end;
if F==3,
    sauv1t=K21;
    K21=1;
    sauv2t=K22;
    K22=0;
    sauv3t=K23;
    K23=0;

    fprintf('1ère partie de RAZ, soyez patient\n');
    init_lab; %fermeture totale des
vannes
    i=1; %pour d'abord faire d
escendre le niveau
    dep_hor1;

```

```

pv1(1)=lire_an(7);
pv3(1)=lire_an(6);
pv2(1)=pv3(1);
while pv1(i)>-9
    i=i+1;
    dep_horl;
    pv1(i)=lire_an(7);
    pv3(i)=lire_an(6);
    pv2(i)=(T*pv3(i)+20*pv2(i-1))/(T+20);
    ecr_na(0,-10);
    ecr_na(1,-10);
    u1(i)=-10;
    u2(i)=-10;
    att_horl (1000*T);
end
i0=i;
pv11(max(size(pv11)+1):(max(size(pv11))+max(size(pv1)
))=pv1;
pv21(max(size(pv21)+1):(max(size(pv21))+max(size(pv2)
))=pv2;
comm1(max(size(comm1)+1):(max(size(comm1))+max(size(u
1))))=u1;
comm2(max(size(comm2)+1):(max(size(comm2))+max(size(u
2))))=u2;
clear pv1;
clear pv2;
clear u1;
clear u2;
init_lab;
e d'eau froide %ouverture de la vann
i=1; %pour ramener la temp
érature à 2
dep_horl;
pv1(1)=lire_an(7);
pv3(1)=lire_an(6);
pv2(1)=pv3(1);
while pv2(i)>2.3
    i=i+1;
    dep_horl;
    pv1(i)=lire_an(7);
    pv3(i)=lire_an(6);
    pv2(i)=(T*pv3(i)+20*pv2(i-1))/(T+20);
    ecr_na(0,-10);
    ecr_na(1,5);
    u1(i)=-10;
    u2(i)=5;
    att_horl (1000*T);
end

```

```

        i1=i;
        pv11(max(size(pv11)+1):(max(size(pv11))+max(size(pv1)
    )))=pv1;
        pv21(max(size(pv21)+1):(max(size(pv21))+max(size(pv2)
    )))=pv2;
        comm1(max(size(comm1)+1):(max(size(comm1))+max(size(u
    1))))=u1;
        comm2(max(size(comm2)+1):(max(size(comm2))+max(size(u
    2))))=u2;
        fprintf('2ème partie de RAZ\n');
        nb_pts=2500-(i0+i1);
    end

```

```

%*****

```

```

pintauto;

```

```

%*****

```

```

K11=sauv1n;
K12=sauv2n;
K13=sauv3n;
if (F==3),
    K21=sauv1t;
    K22=sauv2t;
    K23=sauv3t;
end;

if Fb==2,
    F=Fb;
end;

w1=u1(nb_pts)+w1;      % Détermination des nouveaux offsets
w2=u2(nb_pts)+w2;      % sur les commandes des vannes

retour=0;

clear pv1;
clear pv2;
clear u1;
clear u2;

```

```

%=====
% Programme Reglak3.m
% SOUS-PROGRAMME DU PROGRAMME PINPRI2.m
% INTERVENANT SANS LA PHASE DE RÉGLAGE DE k3
% VERSION:1 JUIN 1997 PAR EC/CB/RMDS
%=====
overshootn=0;
overshoott=0;

while ((overshootn<ampln3)&(overshoott<amplt3)),

    if ((Fr==1)|(Fr==3))
        K13=K13+incK13;
    end

    if ((Fr==2)|(Fr==3))
        K23=K23+incK23;
    end

    ref1=2;
    ref2=5;
    nb_pts=1500;
    if F==3
        nb_pts=2000;
    end

    fprintf('\nEXPÉRIENCE %g.\n',z+z0)
    fprintf ('K13= %g.\t', K13);
    fprintf ('K23= %g.\n', K23);

    pintauto

    if ((F==1)|(F==3)),

        pos=pv1;
        rech_osc;
        osciln=oscil;

        overshootn=ampln3+1;
        if osciln==1,
            fprintf('sur le niveau\n');
        else
            maxin=max(pv1);
            statn=mean(pv1(0.7*max(size(pv1)):0.9
*max(size(pv1)))));
            overshootn=(maxin-statn)/statn;
            depassement3n(z)=overshootn;
            fprintf('\ndépassement niveau= %g.\n'

```

```

,overshootn);
    end;
end;

if ((F==2) | (F==3)),

    pos=pv2-pv2(1);
    rech_osc;
    oscilt=oscil;

    overshoott=amplt3+1;
    if oscilt==1,
        fprintf('sur la température\n');
    else
        maxit=max(pv2);
        statt=mean(pv2(0.7*max(size(pv2)):0.9
*max(size(pv2)))));
        overshoott=(maxit-statt)/(statt-pv2(1
));

        depassement3t(z)=overshoott;
        fprintf('\ndépassement température= %
g.\n',overshoott);
    end;
end;

if F==1,
    subplot(224),plot(pv1);grid;
    subplot(211),plot(pv11); grid;
end;

if F==2,
    subplot(224),plot(pv2);grid;
    subplot(211),plot(pv21); grid;
end;

if F==3,
    subplot(211),plot(pv11); grid;
    subplot(212),plot(pv21); grid;
end;

pause(5);

raz;

if F==1,
    subplot(211),plot(pv11); grid;
end;

```

```
        if F==2,
            subplot(211),plot(pv21); grid;
        end;

        if F==3,
            subplot(211),plot(pv11); grid;
            subplot(212),plot(pv21); grid;
        end;

        pause(5);

        z=z+1;
    end;
```

```

% =====
% NEWGAIN.M
% SUB PROGRAM OF PINTPRI2.m
% which computes from current PD gains
% and associated results, a new set of
% alfa_1, alfa_2 and k1, k2 gains that are susceptible
% to improve system response.
% Version: 27 may 1997 by RMDS/EC/CB
% =====

N=max(size(pos)); % (N=number of response samples)
tt=1:N;           % (tt=time scale)
T=tech;           % (T= sampling period)
N1=.7*N;
N2=.9*N;

mx_1=max(pos(N1:N2));
mn=min(pos(N1:N2));
srnawrt=0;
if (mx_1-mn)/(mx_1+mn)>.08,
    input('stationary regime not achieved within required
    time');
    srnawrt=1;
end;

% verification that stationary regime has
indeed
% be attained N1*T,N2*T interval

if srnawrt==0
    stat_v=mean(pos(N1:N2));
    mx_2=max(pos(3:N1)); % (peak value of response)
    overshoot=(mx_2-stat_v)/stat_v;

% =====
% = We compute values of alfa_1, alfa_2, and wn,
% = associated with current test results. We do this by
% = evaluating the best second order approximant
% =====
% (uses ordre_2.m)
% =====
% procedure: 1: rough evaluation of alfa_1, alfa_2, ksi, wn
% ===== from collected data
%           2: use of rough values of alfa_1, alfa_2
%           to determine alfa_1, alfa_2 of a 2nd order sy
stem
%           with same overshoot and same time_of_first_pe
ak

```


%

as the test_bench system

```

if overshoot<=.05
    ksi=1;
    i=1;

    while pos(i)<=.9*stat_v,
        i=i+1;
    end;

    trn=i*T;
    wn=4/trn;
    alfa_1=2*ksi/wn;
    alfa_2=1/wn^2;
    tfpa=trn;
    ordre_2;

else
    temp=-log(overshoot)/pi;
    ksi=sqrt(temp^2/(1+temp^2));
    i=1;

    while pos(i)~=mx_2,
        i=i+1;
    end;

    tfp=i*T; % (time first peak)
    wn=pi/(tfp*sqrt(1-ksi^2));
    alfa_1=2*ksi/wn;
    alfa_2=1/wn^2;

    ordre_2;

    i=1;
    mx_3=max(y_alfa(1:N));

    while (mx_3-y_alfa(i))>0.000001,
        i=i+1;
    end;

    tfpa=i*T; % (time first peak)

    wn=(tfpa/tfp)*wn;
    alfa_1=2*ksi/wn;
    alfa_2=1/wn^2;
end;

```

```

% =====
% = From current values of k1, k2 and alfa_1, alfa_2,
% = and wn, we compute new values of k1, k2. These values
% = are computed so as to have ksi=ksid and wn = mu*wn0.
% =====

% Old values

    k10=k1;
    k20=k2;
    alfa_10=alfa_1;
    alfa_20=alfa_2;
    wn0=wn;
    ksi0=ksi;

% New values

    if F==1,
        inst=instn;
        ampl=ampln12;
    end;
    if F==2,
        inst=instt;
        ampl=amplt12;
    end;

    mu=1.1;
    if(tfpa<inst),
        mu=1;
    end;

    muk=1.2;
    if(overshoot<ampl),
        muk=1;
    end;

    wn=mu*wn0;
    ksid=muk*ksi0;

    alfa_1=2*ksid/wn;
    alfa_2=1/wn^2;

    k1 = k10*alfa_20/alfa_2;
    k2 = k20 + k1*alfa_1-k10*alfa_10;

end

```

```

%=====
% Programme Ordre 2.m
% SOUS PROGRAMME DU PROGRAMME NEWGAIN.m
%=====
num2=[1];
den2=[alfa_2 alfa_1 1];
[a2,b2,c2,d2]=tf2ss(num2,den2);
[a2d,b2d]=c2d(a2,b2,T);
x_alfa(:,1)=[0 0]';
for i=1:N-1,
    x_alfa(:,i+1)=a2d*x_alfa(:,i)+b2d*stat_v;
end;
y_alfa=c2*x_alfa;

```

```

% *****
% Programme: Pint_eds.m
% Ce programme a été écrit avec MATLAB 4.2c. Il permet d'im-
% planter un ou deux régulateurs PID Dual Loop pour l'asser-
% ssement en et/ou en température de l'eau dans le réservoir
% du banc d'essai de la section automatique de l'École
% Polytechnique de Montréal. Les données en entrée sont
% essentiellement les mesures du niveau et de la% température
% fournies par une carte d'entrées/sorties Labmaster
% (canaux 7 et 6). Les sorties sont les commandes envoyées
% sur la vanne d'eau froide et la vanne d'eau chaude par
% l'intermédiaire de la même carte Labmaster (canaux 0 et 1).
% Les variables que l'utilisateur peut modifier sont :
% - F (indicateur de la ou des boucles à asservir)
% - les gains des régulateurs
% - la durée de l'expérience
% - la durée du retard introduit sur les commandes
% - la valeur de la perturbation ajoutée au milieu de l'expé-
% rience sur la vanne d'eau froide pour tester la robuste-
% sse de l'asservissement de niveau seul ou de l'ensemble
% asservissement de niveau/asservissement de température
% sur la vanne d'eau froide et sur la vanne d'eau chaude
% pour tester la robustesse de l'asservissement de tempéra-
% ture seule.
% *****

fprintf('*****\n');

fprintf('
                                Compensateur PID Dual-Loop
\n');
fprintf('
                                Asservissement de niveau et température.
\n');
fprintf('Version: 10/06/97. par: EC/RMDS/CB
\n');

fprintf('*****\n');

fprintf('*****\n')
fprintf('
                                Données par défaut:  \n ');

fprintf('*****\n')

```

```

% INITIALISATION

clear

    echo on

    nb_pts=100; % nombre de points de l'expérience
    T=0.5;      % temps d'échantillonnage

% Paramètres du régulateur PID Dual Loop

    K11 = 1.02;
    K12 = 18;
    K13 = 0.003;
    K14 = 0;
    A11 = 50.8;
    A12 = 1455;

    ref1 =2;
    K21 = 1.82;
    K22 = 44.5;
    K23 = 0.009;
    K24 = 0;
    A21 = 72.3;
    A22 = 1305;

    ref2=5;      % consigne désirée
    alpha1=1.723; % rapport d'eau froide/eau chaude

    echo off

F=input('Asservissement de: 1.-Niveau 2.-Température 3.-Niveau et Température=');
P=input(' Échelon de perturbation sur le niveau/température: 1.-oui 0.-non=');
ret=input(' Retard sur le niveau/température, 1.-oui, 0.-non=');

if ret==1,
    retard=input('valeur de retard*0.5 en secondes=');
end

```

```
% OPPORTUNITÉ DE CHANGEMENT
```

```
fprintf('*****\n');
```

```
fprintf(' Si vous voulez changer les données, tapez maintenant \n ')
fprintf(' les variables désirées et tapez break, enter.\n')
fprintf(' Sinon pour lancer l"expérience, tapez break, enter.\n ')
fprintf('*****\n')
```

```
keyboard
```

```
% Conditions initiales d'opération
```

```
alpha11=1/(1+alpha1);
alpha12=alpha1*alpha11;
```

```
alpha21=0.36;
alpha22=-0.64;
```

```
u11 = zeros(1,nb_pts);
u12 = zeros(1,nb_pts);
u21 =zeros(1,nb_pts);
u22 = zeros(1,nb_pts);
pv1 = zeros(1,nb_pts);
pv2 = zeros(1,nb_pts);
pvv2 =zeros(1,nb_pts);
err1 = zeros(1,nb_pts);
err2 = zeros(1,nb_pts);
int_err1= zeros(1,nb_pts);
der1=zeros(1,nb_pts);
der2=zeros(1,nb_pts);
int_err2= zeros(1,nb_pts);
U1 = zeros(1,nb_pts);
U2 = zeros(1,nb_pts);
```

```
if F==1,
    alpha21=0;
    alpha22=0;
end
```

```
if F==2,
    alpha11=0;
```

```

        alpha12=0;
end

fprintf('SOYEZ PATIENT!!!!\n');
est=0;
fprintf('---1---2---3---4---5---6---7---8---9---10\n');

        init_lab;% INITIALISATION CONVERTISSEUR

% BOUCLE DE REGULATION

pert=0;

dep_hor1;
pv1(1)=lire_an(7);
ref1=ref1-pv1(1);
consigne1=ref1*ones(1,nb_pts);

pv3(1)=lire_an(6);
pv2(1)=pv3(1);
ref2=ref2-pv2(1);
consigne2=ref2*ones(1,nb_pts);

for i=2:nb_pts,

        dep_hor1;

        if (100*i/nb_pts)>(est+2.45) est=est+2.5; fprintf('*');
        end

        pv1(i) = lire_an(7);
        pv3(i) = lire_an(6);

% PERTURBATION pert

        if P==1,
                if i>=nb_pts/2;
                        pert=1;
                end
        end

% Commande U1, niveau

        pvv1(i)=pv1(i)-pv1(1);
        err1(i) = consigne1(1,i)-pvv1(i);
        int_err1(i)=int_err1(i-1)+err1(i)*T;
        der1(i)=(pvv1(i)-pvv1(i-1))/T;
        U1(i)=K11*err1(i) + K13*(int_err1(i)-A11*pvv1(i)-A12*

```

```

der1(i))+K14*ref1-K12*der1(i);

% Filtre numérique pour la température:

    pv2(i)=(T*pv3(i)+20*pv2(i-1))/(T+20);

% commande U2, température:

    pvv2(i)=pv2(i)-pv2(1);
    err2(i) = consigne2(1,i)-pvv2(i);
    der2(i) = (pvv2(i)-pvv2(i-1))/T;
    int_err2(i) = int_err2(i-1)+err2(i)*T;
    U2(i)=K21*err2(i) + K23*(int_err2(i) - A21*pvv2(i)-A2
2*der2(i))+K24*ref2-K22*der2(i);

% Conditions de retard, pour toutes les boucles:

    if F==1,
        if ret==1,
            if i<=retard,
                UU1(i)=0;
            elseif i>retard,
                UU1(i)=U1(i-retard);
            end
            u11(i)=alpha11*UU1(i);
            u12(i)=alpha12*UU1(i)+pert;
            u21(i)=0;
            u22(i)=0;
        end
        if ret==0,
            u11(i)=alpha11*U1(i);
            u12(i)=alpha12*U1(i)+pert;
            u21(i)=0;
            u22(i)=0;
        end
    end

    if F==2
        if ret==1,
            if i<=retard,
                UU2(i)=0;
            elseif i>retard,
                UU2(i)=U2(i-retard);
            end

```



```

        u21(i)=alpha21*(UU2(i)+pert);
        u22(i)=alpha22*(UU2(i)+pert);
    end

    if ret==0,
        u21(i)=alpha21*(U2(i)+pert);
        u22(i)=alpha22*(U2(i)+pert);
        u12(i)=0;
        u11(i)=0;
    end
end

if F==3,
    if ret==1,
        if i<=retard,
            UU1(i)=0;
            UU2(i)=0;
        elseif i>retard,
            UU1(i)=U1(i-retard);
            UU2(i)=U2(i-retard);
        end
        u11(i)=alpha11*UU1(i);
        u12(i)=alpha12*UU1(i)+pert;
        u21(i)=alpha21*UU2(i);
        u22(i)=alpha22*UU2(i);
    end

    if ret==0,
        u11(i)=alpha11*U1(i);
        u12(i)=alpha12*U1(i)+pert;
        u21(i)=alpha21*U2(i);
        u22(i)=alpha22*U2(i);
    end
end

end

% commandes avec prédécouplage

u1(i) = u11(i) + u21(i);
u2(i) = u12(i) + u22(i);

% signaux de commandes envoyés aux interfaces pour chaque van
ne

ecr_na (0, u1(i));
ecr_na (1, u2(i));

att_hor1 (1000 * T);

```

```

end;

    ecr_na (0, 0);
    ecr_na (1, 0);

% AFFICHAGE DE LA REPONSE SELON LE CHOIX DE L'ASSERVISSEMENT
:
temps = [1:1:nb_pts] * T;
if F==1,
    subplot(211)
    plot(temps,pv1,'r')
    title('Comportement du niveau')
    xlabel('temps en secondes')
    ylabel('valeurs en volts')
    grid

    subplot(212)
    plot(temps, u1,'g', temps,u2,'y')
    title('Comportement de la commande u1 et u2')
    xlabel('temps en secondes')
    ylabel('valeurs en volts')
    grid
end

if F==2,
    subplot(211)
    plot(temps,pv2,'r')
    title('Comportement de la temperature')
    xlabel('temps en secondes')
    ylabel('valeurs en volts')
    grid

    subplot(212)
    plot(temps,u1,'g',temps,u2,'y')
    title('Comportement de la commande u1 et u2')
    xlabel('temps en secondes');
    ylabel('valeurs en volts')
    grid
end

if F==3,
    subplot(221)
    plot(temps,pv1,'r')
    title('Comportement du niveau')
    xlabel('temps en secondes')

```

```

        ylabel('valeurs en volts')
        grid
        subplot(222)
        plot(temps,u1,'y')
        title('commande vanne d"eau chaude')
        xlabel('temps en secondes')
        ylabel('valeurs en volts')
        grid
        subplot(223)
        plot(temps,pv2,'g')
        title('Comportement de la temperature')
        xlabel('Temps en secondes')
        ylabel('valeurs en volts')
        grid

        subplot(224)
        plot(temps,u2,'w')
        title('commande vanne d"eau froide')
        xlabel('Temps en secondes')
        ylabel('valeurs en volts')
        grid
    end

% CHOIX DE SAUVERGADER LES DONNÉES DE L'EXPERIENCE

E=input('Vous voulez enregistrer les données(o/n):','s');
if E=='o',
    nom='a:\test1'
    keyboard
    eval(['save ' nom ' pv1 pv2 u11 u12 u21 u22 U1 U2 tem
ps K11 K12 K13 K14 A11 A12 K21 K22 K23 K24 A21 A22 T ref1 ref
2']));
end

```

```

% Programa rn.m
% Ce programme utilise la structure d'un neuro-compensateur e
t d'un neuro-emulateur en utilisant
% les notions presenté par M. Omatu (1996). Ce programme util
ise les fichiers: weights.mat, tsig.m
% Le programme rn.m est utilisé pour l'asservissement du nive
au et de la température.
% Date: 12/08/1997 par: Angel Ruiz et Ernesto Cornieles

% INITIALISATION

clear

% Parametres du controleur
% =====
clear

    echo on

    nb_pts=4000;
    T=0.5;

    ref1=2;
    ref2=4;
    echo off

F=input('Asservissement de 1.-Niveau 2.-Température 3.-Niveau
et temperature=');
P=input('Echelon de perturbation sur le niveau/température: 1
.-oui 0.-nom=');
ret=input('Retard sur le niveau/température, 1.-oui, 0.-nom='
);

if ret==1,
retard=input('valeur de retard*0.5 en secondes=');
end

% OPPORTUNITE DE CHANGEMENT
fprintf('*****\n');

fprintf(' Si vous voulez changer les donnees, taper mainten
t\n')
fprintf(' les variables desirees et taper break, enter.\n')
fprintf(' Pour demarrer l''experience taper break, enter.\n'
)

```

```

fprintf('*****\n')

keyboard

% Conditions d'operation
    alpha1=2;
    alpha11=1/(1+alpha1);
    alpha12=alpha1*alpha11;
    alpha21=0.36;
    alpha22=-.64;

    pv1 = zeros(1,nb_pts);
    pv2 = zeros(1,nb_pts);

    U1 = zeros(1,nb_pts);
    U2 = zeros(1,nb_pts);

    consigne1 = ref1*ones(1,nb_pts);
    consigne2 = ref2*ones(1,nb_pts);

    pert=0;

fprintf('SOYEZ PATIENT!!!!\n');
est=0;
fprintf('---1---2---3---4---5---6---7---8---9---0\n');
    init_lab;          % INITIALISATION CONVERTISSEUR

% BOUCLE DE REGULATION

dep_hor1;
pv3(1)=lire_an(6);
pv2(1)=pv3(1);

load wnt

for i=2:nb_pts,

    dep_hor1;

    if (100*i/nb_pts)>(est+2.45) est=est+2.5; fprintf('*'
);
    end

    pv1(i) = lire_an(7);

```

```

pv3(i) = lire_an(6);

% Filtre numerique pour la temperature

pv2(i)=(T*pv3(i)+20*pv2(i-1))/(T+20);

% Creation du vecteur v = r(t) y(t-10) y(t-20) u(t-10)

v3=0;
v4=0;
v5=0;
if i>4,
v3=pv1(i-4);
v5=U1(i-4);
end
if i>7
v4=pv1(i-7);
end

if F==1 | F==3,

v1=consigne1(i);
v2=pv1(i);
v=[v1;v2;v3;v4;v5];

%*****
%*****

% Fonction complete pour Nc=3 couches et Ne=2 couches
% Recoit le vecteur v(5x1) = r(t) y(t) y(t-10) y(t-20) u(t-10)
)
% Update des poids du Nc puis Ne avec les trois derniers echa
ntillons.
% Retourne la prochaine valeur de commande.
% Utilise le fichier weights.mat avec W1-W5 et B1-B5 plus les
vecteurs p et t.
% Entrees pour le Nc : r(t) y(t-10) y(t-20) u(t-10)
% Entrees pour le Ne : u(t) y(t-10) y(t-20) u(t-10)

% Creation des vecteurs Xc et Xe

v=v/10;
p=[v(1,1);v(3,1);v(4,1);v(5,1)]; % Nc = r(t) y(t-1) y(
t-2) u(t-1)
t=v(2,1); % Vecteur target t =

```

```

y(t)
u=[0;p(2,:);p(3,:);p(4,:)];          % Ne = a3(t) y(t-1) y
(t-2) u(t-1)

% Boucle update NE

    a1=tsig(W1*p,B1);
    a2=tsig(W2*a1,B2);
    a3=(W3*a2+B3);
    u(1,1)=a3;
    a4=tsig(W4*u,B4);
    a5=W5*a4+B5;

    % Erreur et calcul des deltas

    e=t-a5;                          % Erreur = y(t)-a5(t)
    d5=e;
    d4=(ones-(a4.*a4)).*(W5'*d5);

    % Update des poids du NE

    lr=0.025;
    W5=W5+lr*(d5*a4');
    B5=B5+lr*d5;
    W4=W4+lr*(d4*u');
    B4=B4+lr*d4;

% Boucle update NC

a4=tsig(W4*u,B4);
a5=(W5*a4)+B5;

% Erreur et calcul des deltas

e=(p(1,1)-a5);                      % Erreur = r(t)-a5(t)
d5=e*3;
d4=(ones-(a4.*a4)).*(W5'*d5);
d3=d4'*W4(:,1);
d2=(ones-(a2.*a2)).*(W3'*d3);
d1=(ones-(a1.*a1)).*(W2'*d2);

% Update des poids du Nc

lr =2;
D3=lr*(d3*a2');
W3=W3+D3;

```

```

C3=lr*d3;
B3=B3+C3;
D2=lr*(d2*a1');
W2=W2+D2;
C2=lr*d2;
B2=B2+C2;
D1=lr*(d1*p');
W1=W1+D1;
C1=lr*d1;
B1=B1+C1;

```

```
% Valeurs a appliquer
```

```

a1=tsig(W1*p,B1);
a2=tsig(W2*a1,B2);
a3=(W3*a2+B3);
%if a3>3,
%a3=3;
%end
%if a3<-3,
%a3=-3;
%end

```

```
U1(i)=a3;
```

```
k=.997;
```

```

B1=B1-C1*k;
B2=B2-C2*k;
B3=B3-C3*k;
W1=W1-D1*k;
W2=W2-D2*k;
W3=W3-D3*k;

```

```

v3=0;
v4=0;
v5=0;
if i>7,
v3=pv2(i-7);
v5=U2(i-7);
end
v1=consigne2(i);
v2=pv2(i);
v=[v1;v2;v3;v4;v5];

```

```
% *****
```



```

*****

% Function complete pour Nc=3 couches et Ne=2 couches
% Recoit le vecteur v(5x1) = r(t) y(t) y(t-10) y(t-20) u(t-10)
)
% Update des poids du Nc puis Ne avec les trois derniers echa
ntillons.
% Retourne la prochaine valeur de commande.
% Utilise le fichier weights.mat avec W1-W5 et B1-B5 plus les
vecteurs p et t.
% Entrees pour le Nc : r(t) y(t-10) y(t-20) u(t-10)
% Entrees pour le Ne : u(t) y(t-10) y(t-20) u(t-10)

% Creation des vecteurs Xc et Xe

v=v/10;
q=[v(1,1);v(3,1);v(4,1);v(5,1)];           % Nc = r(t) y(t-1) y(
t-2) u(t-1)
t=v(2,1);                                   % Vecteur target t =
y(t)
u=[0;p(2,:);p(3,:);p(4,:)];                % Ne = a3(t) y(t-1) y
(t-2) u(t-1)

% Boucle update NE

    a6=tsig(W6*q,B6);
    a7=tsig(W7*a6,B7);
    a8=(W8*a7+B8);
    u(1,1)=a8;
    a9=tsig(W9*u,B9);
    a10=W10*a9+B10;

    % Erreur et calcul des deltas

    e=t-a10;                               % Erreur = y(t)-a5(t)
    d10=e;
    d9=(ones-(a9.*a9)).*(W10'*d10);

    % Update des poids du NE

    lr=0.025;

    W10=W10+lr*(d10*a9');
    B10=B10+lr*d10;
    W9=W9+lr*(d9*u');
    B9=B9+lr*d9;

% Boucle update NC

```

```

a9=tsig(W9*u,B9);
a10=(W10*a9)+B10;

% Erreur et calcul des deltas

e=(q(1,1)-a10); % Erreur = r(t)-a5(t)
d10=e*2;
d9=(ones-(a9.*a9)).*(W10'*d10);
d8=d9'*W9(:,1);
d7=(ones-(a7.*a7)).*(W8'*d8);
d6=(ones-(a6.*a6)).*(W7'*d7);

% Update des poids du Nc

lr =2;

D8=lr*(d3*a2');
W8=W8+D8;
C8=lr*d8;
B8=B8+C8;
D7=lr*(d7*a6');
W7=W7+D7;
C7=lr*d7;
B7=B7+C7;
D6=lr*(d6*q');
W6=W6+D6;
C6=lr*d6;
B6=B6+C6;

% Valeurs a appliquer

a6=tsig(W6*q,B6);
a7=tsig(W7*a6,B7);
a8=(W8*a7+B8);

if a8>5.5,
a8=5.5;
end
if a8<-5.5,
a8=-5.5;
end
U2(i)=a8;

% Sauvegarde des valeurs

k=.997;

```

```

B6=B6-C6*k;
B7=B7-C7*k;
B8=B8-C8*k;
W6=W6-D6*k;
W7=W7-D7*k;
W8=W8-D8*k;

end

%**** Boucle de perturbation ****

if P==1 & i>=nb_pts/2;
pert=1;
end

%**** Boucle de retard ****

if F==1,
u11(i)=alpha11*U1(i);
u12(i)=alpha12*U1(i)+pert;
UU1(i)=0;

    if ret==1,
        if i>retard,
            UU1(i)=U1(i-retard);
        end
        u11(i)=alpha11*UU1(i);
        u12(i)=alpha12*UU1(i)+pert;
    end

u21(i)=0;
u22(i)=0;
end

if F==2,
u21(i)=alpha21*(U2(i)+pert);
u22(i)=alpha22*(U2(i)+pert);
UU2(i)=0;

    if ret==1
        if i>retard,
            UU2(i)=U2(i-retard);
        end
        u21(i)=alpha21*(UU2(i)+pert);
        u22(i)=alpha22*(UU2(i)+pert);
    end
end

```

```

u12(i)=0;
u11(i)=0;

end

if F==3,
u21(i)=alpha21*U2(i);
u22(i)=alpha22*U2(i);
u11(i)=alpha11*U1(i);
u12(i)=alpha12*U1(i)+pert;
UU1(i)=0;
UU2(i)=0;

if ret==1
    if i>retard
        UU1(i)=U1(i-retard);
        UU2(i)=U2(i-retard);
    end
    u21(i)=alpha21*UU2(i);
    u22(i)=alpha22*UU2(i);
    u11(i)=alpha11*UU1(i);
    u12(i)=alpha12*UU1(i)+pert;
end
end
%**** Ecriture ****

u1=u11+u21;
u2=u12+u22;

ecr_na (0, u1(i));
ecr_na (1, u2(i));

att_horl (1000 * T);

end

ecr_na (0, 0);
ecr_na (1, 0);

% AFFICHAGE REPONSE:

temps = [1:1:nb_pts] * T;
subplot(221),plot(temps,pv1)
title('Comportement du niveau')
xlabel('Temps en secondes')
ylabel('Volts')
grid

```

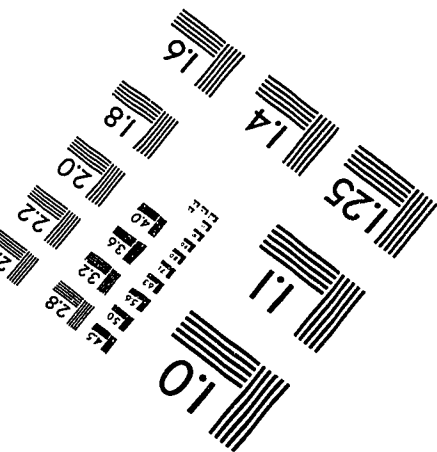
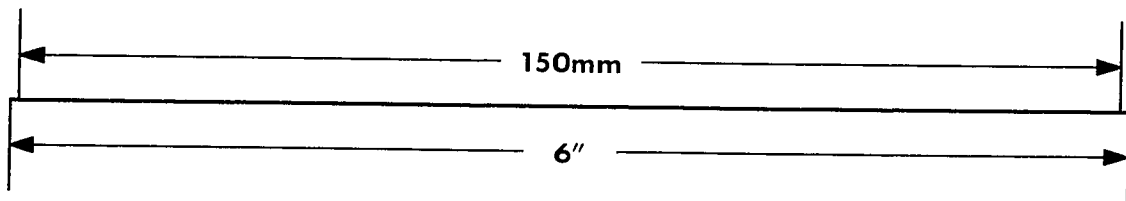
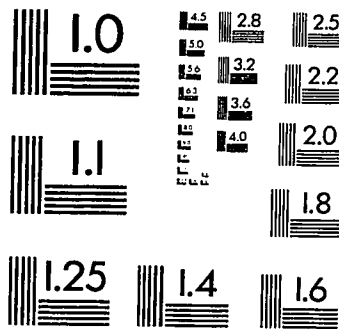
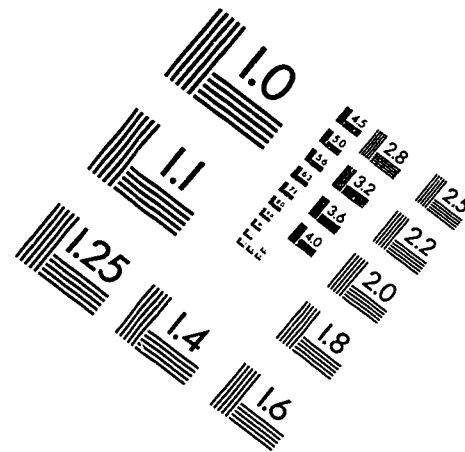
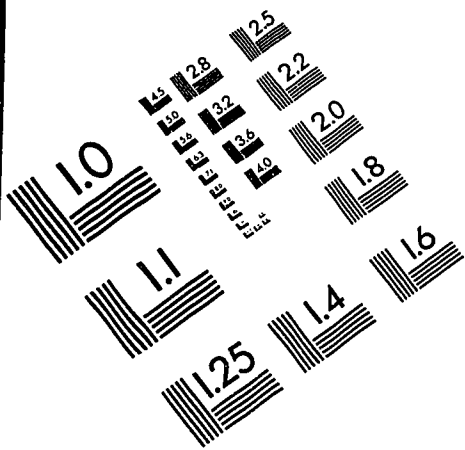
```

subplot(222),plot(temps,u1)
title('Commmmande d"eau chaude')
xlabel('Temps en secondes')
ylabel('Volts')
grid
subplot(223),plot(temps,pv2)
title('Comportement de la temperature')
xlabel('Temps en secondes')
ylabel('Volts')
grid
subplot(224),plot(temps,u2)
title('Commande d"eau froide')
xlabel('temps(secondes)')
ylabel('Volts')
grid
E=input('Vous voulez enregistrer les donnees(o/n):','s');
if E=='o',
nom='a:\test1'
keyboard
eval(['save ' nom ' pv1 pv2  u1 u2 temps ']);
end

E=input('voulez-vous enregistrer les poids(o/n):','s');
if E=='o',
save weights B1 B2 B3 B4 B5 B6 B7 B8 B9 B10 W1 W2 W3 W4 W5 W6
W7 W8 W9 W10
end

```

IMAGE EVALUATION TEST TARGET (QA-3)



APPLIED IMAGE, Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989

© 1993, Applied Image, Inc., All Rights Reserved

